

Sistema de Gestión Académica

Documentación Completa del Proyecto PHP

Total de archivos: 100

Índice de Archivos

#	Archivo	Ruta
1	ARCHITECTURE.md	/
2	FuncionalidadesRol.md	/
3	README.md	/
4	ResumenEjecutivo_08C.md	/
5	SugerenciasFuturas.md	/
6	mallá.md	/
7	men#U00fa_v2.0.md	/
8	.htaccess	ArchImp/config
9	config.php	ArchImp/config
10	database.php	ArchImp/config
11	env.php	ArchImp/config
12	AcademicController.php	ArchImp/controllers
13	AssignmentController.php	ArchImp/controllers
14	AttendanceController.php	ArchImp/controllers
15	AuthController.php	ArchImp/controllers
16	BackupController.php	ArchImp/controllers
17	DashboardController.php	ArchImp/controllers
18	ExportController.php	ArchImp/controllers
19	InstitutionController.php	ArchImp/controllers
20	JustificationController.php	ArchImp/controllers
21	NotificationController.php	ArchImp/controllers
22	ProfileController.php	ArchImp/controllers
23	ReportController.php	ArchImp/controllers
24	ReportController2.php	ArchImp/controllers
25	RepresentativeController.php	ArchImp/controllers
26	ScheduleController.php	ArchImp/controllers
27	SearchController.php	ArchImp/controllers
28	StatsController.php	ArchImp/controllers
29	TutorController.php	ArchImp/controllers
30	UserController.php	ArchImp/controllers
31	Backup.php	ArchImp/helpers
32	Logger.php	ArchImp/helpers
33	Mailer.php	ArchImp/helpers
34	Security.php	ArchImp/helpers
35	Attendance.php	ArchImp/models
36	ClassSchedule.php	ArchImp/models
37	Course.php	ArchImp/models

38	Institution.php	ArchImp/models
39	InstitutionShift.php	ArchImp/models
40	Justification.php	ArchImp/models
41	Notification.php	ArchImp/models
42	Representative.php	ArchImp/models
43	Role.php	ArchImp/models
44	SchoolYear.php	ArchImp/models
45	Shift.php	ArchImp/models
46	Subject.php	ArchImp/models
47	TeacherAssignment.php	ArchImp/models
48	User.php	ArchImp/models
49	.htaccess	ArchImp/public
50	img.php	ArchImp/public
51	index.php	ArchImp/public
52	global.css	ArchImp/public/css
53	course_edit.php	ArchImp/views/academic
54	course_students.php	ArchImp/views/academic
55	course_subject_edit.php	ArchImp/views/academic
56	course_subjects.php	ArchImp/views/academic
57	create.php	ArchImp/views/academic
58	edit.php	ArchImp/views/academic
59	enroll.php	ArchImp/views/academic
60	index.php	ArchImp/views/academic
61	school_year_create.php	ArchImp/views/academic
62	school_year_edit.php	ArchImp/views/academic
63	subject_edit.php	ArchImp/views/academic
64	index.php	ArchImp/views/assignments
65	tutor.php	ArchImp/views/assignments
66	view_course.php	ArchImp/views/assignments
67	calendar.php	ArchImp/views/attendance
68	course_view.php	ArchImp/views/attendance
69	my_attendance.php	ArchImp/views/attendance
70	register.php	ArchImp/views/attendance
71	view.php	ArchImp/views/attendance
72	forgot.php	ArchImp/views/auth
73	login.php	ArchImp/views/auth
74	register.php	ArchImp/views/auth
75	reset.php	ArchImp/views/auth
76	index.php	ArchImp/views/dashboard

77	index.php	ArchImp/views/institution
78	my_list.php	ArchImp/views/justifications
79	pending.php	ArchImp/views/justifications
80	reviewed.php	ArchImp/views/justifications
81	submit.php	ArchImp/views/justifications
82	index.php	ArchImp/views/notifications
83	navbar.php	ArchImp/views/partials
84	change_password.php	ArchImp/views/profile
85	edit.php	ArchImp/views/profile
86	view.php	ArchImp/views/profile
87	index.php	ArchImp/views/reports
88	child_attendance.php	ArchImp/views/representatives
89	manage.php	ArchImp/views/representatives
90	my_children.php	ArchImp/views/representatives
91	index.php	ArchImp/views/schedule
92	manage.php	ArchImp/views/schedule
93	index.php	ArchImp/views/stats
94	course_attendance.php	ArchImp/views/tutor
95	dashboard.php	ArchImp/views/tutor
96	no_tutor.php	ArchImp/views/tutor
97	search_students.php	ArchImp/views/tutor
98	create.php	ArchImp/views/users
99	edit.php	ArchImp/views/users
100	index.php	ArchImp/views/users

Estructura de Carpetas

```
■■■ ArchImp
■ ■■■ 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
■ ■ ■■■ css
■ ■ ■ ■■■ global.css
■ ■ ■■■ .htaccess
■ ■ ■■■ img.php
■ ■ ■■■ index.php
```

- ■■■ views
- ■■■ academic
 - ■■■■ course_edit.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
- ■■■ 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
- ARCHITECTURE.md
- FuncionalidadesRol.md
- README.md
- ResumenEjecutivo_08C.md
- SugerenciasFuturas.md
- malla.md
- men#U00fa_v2.0.md

Código Fuente

1. ARCHITECTURE.md

```
1 Tu sistema tiene estructura tipo MVC clásico en PHP (sin framework pesado), así que para darle continuidad sin perder la.
2
3 ■ Flujo del sistema
4
5 ■ Seguridad
6
7 ■■ Acceso a datos
8
9 ■ Lógica de negocio
10
11 Te lo explico estratégicamente.
12
13 ■ 1. El CEREBRO del sistema (Lo más importante)
14 ■ /config
15
16 Aquí está la base del sistema.
17
18 Archivos críticos:
19
20 database.php → conexión a BD
21
22 config.php → constantes globales
23
24 posibles rutas base, URL_BASE, etc.
25
26 ■ Si esto falla, nada funciona.
27 ■ Aquí defines entorno (local / producción).
28
29 ■ /controllers ■ (Muy importante)
30
31 Este es el flujo principal del sistema.
32
33 Aquí están:
34
35 AttendanceController.php
36
37 UserController.php
38
39 AuthController.php
40
41 ReportController.php
42
43 JustificationController.php
44
45 etc.
46
47 ■ Aquí debes mirar:
48
49 Cómo se reciben $_GET['action']
50
51 Cómo se enrutan las vistas
52
53 Cómo se llaman los modelos
54
55 ■ Si quieres entender el sistema, empieza leyendo:
56
```



```
57 public/index.php
58
59 Luego el archivo principal que maneje el router
60
61 Después los controllers
62
63 ■ 2. La LÓGICA del sistema
64 ■ /models ■
65
66 Aquí está la lógica que habla con la base de datos.
67
68 Ejemplos:
69
70 User.php
71
72 Attendance.php
73
74 Course.php
75
76 Justification.php
77
78 Aquí debes revisar:
79
80 Consultas SQL
81
82 Métodos como:
83
84 getById()
85
86 create()
87
88 update()
89
90 delete()
91
92 getByStudent()
93
94 ■ Si quieres ampliar el sistema sin romperlo, este es el lugar más delicado.
95
96 ■■ 3. La BASE DE DATOS
97 ■ /database
98
99 Aquí normalmente encuentras:
100
101 Script SQL inicial
102
103 Migraciones
104
105 Estructura de tablas
106
107 ■ Muy importante revisar:
108
109 Relaciones
110
111 Foreign Keys
112
113 Índices
114
115 Campos ENUM (como status: presente, ausente, justificado)
```

116
117 Si no entiendes la BD, no podrás escalar el sistema correctamente.
118
119 ■■ 4. Las VISTAS (UI)
120 ■ /views
121
122 Están organizadas por módulos:
123
124 attendance
125
126 users
127
128 reports
129
130 justifications
131
132 dashboard
133
134 etc.
135
136 ■ Aquí solo hay presentación (HTML + PHP básico).
137 ■ No debería haber lógica compleja.
138
139 ■■ Si encuentras consultas SQL aquí → mala práctica.
140
141 ■ 5. Seguridad
142 ■ /helpers
143
144 Muy importante revisar:
145
146 Security.php
147
148 funciones de validación
149
150 control de sesiones
151
152 permisos por rol
153
154 Ejemplo:
155
156 Security::requireLogin();
157 Security::requireRole('admin');
158
159
160 Si vas a crecer el sistema (representantes, docentes, estudiantes), esto es CLAVE.
161
162 ■ 6. Punto de Entrada
163 ■ /public
164
165 Aquí normalmente está:
166
167 index.php
168
169 assets
170
171 js
172
173 css
174

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

176

177 Si quieres entender el flujo completo:

178

179 index.php

180

181 router

182

183 controller

184

185 model

186

187 view

188

189 ■ 7. Lo que NO debes tocar mucho

190 ■ /vendor

191

192 Esto es Composer.

193

194 PHPMailer

195

196 PhpSpreadsheet

197

198 TCPDF

199

200 ZipStream

201

202 ■ No modifiques nada aquí.

203 ■ Solo se actualiza con composer.

204

205 ■ 8. uploads

206

207 Aquí guardas:

208

209 justificativos

210

211 archivos PDF

212

213 evidencias

214

215 Debes cuidar:

216

217 Validación de tipo de archivo

218

219 Seguridad (evitar ejecución .php)

220

221 Tamaño máximo

222

223 ■ Si quieres continuidad SIN perder perspectiva

224

225 Te recomiendo este orden mental:

226

227 ■ 1■■ Entender el flujo completo

228

229 Abre:

230

231 public/index.php

232

233

234 Y sigue el recorrido hasta una vista.
235
236 Haz un mapa mental:
237
238 Request → Controller → Model → View
239
240 ■ 2■■■ Documentar la base de datos
241
242 Haz un esquema visual:
243
244 users
245
246 roles
247
248 courses
249
250 attendance
251
252 justifications
253
254 representatives
255
256 Si entiendes la BD, entiendes el 70% del sistema.
257
258 ■ 3■■■ Definir módulos del sistema
259
260 Tu sistema tiene módulos claros:
261
262 ■ Autenticación
263
264 ■■■■ Usuarios
265
266 ■ Académico
267
268 ■ Asistencia
269
270 ■ Justificaciones
271
272 ■ Reportes
273
274 ■ Backups
275
276 Documenta qué hace cada uno.
277
278 ■ Resumen Estratégico
279
280 Si solo pudieras revisar 6 cosas importantes:
281
282 ■ public/index.php
283
284 ■ config/database.php
285
286 ■ controllers (especialmente Attendance y Auth)
287
288 ■ models (Attendance, User, Justification)
289
290 ■ helpers/Security.php
291
292 ■ estructura de la base de datos

293

294 ■ Consejo Profesional

295

296 Para no perder perspectiva:

297

298 ✓■ Crea un archivo ARCHITECTURE.md

299 ✓■ Dibuja el flujo

300 ✓■ Define responsabilidades por carpeta

301 ✓■ No mezcles lógica en vistas

2. FuncionalidadesRol.md

```
1  Quiero implementar estas funciones al sistema
2  Rol: AUTORIDAD
3  Sección: Gestión de Usuarios
4  Funcionalidad: Crear/editar/eliminar usuarios
5
6  # ■ FUNCIONALIDADES POR ROL - ECUASIST 2026
7
8  ## Versión: 1.5
9  **Fecha:** 16 de Febrero de 2026
10 **Cobertura:** 97.4%
11
12 ---
13
14 ## ■ AUTORIDAD
15
16 ### Gestión de Usuarios
17 | # | Funcionalidad | Estado |
18 |---|-----|-----|
19 | 1 | Crear/editar/eliminar usuarios | ■ Implementado |
20 | 2 | Asignar/quitar roles a usuarios | ■ Implementado |
21
22 ### Gestión Académica
23 | # | Funcionalidad | Estado |
24 |---|-----|-----|
25 | 3 | Crear/editar/eliminar años lectivos | ■ Implementado |
26 | 4 | Activar/desactivar año lectivo | ■ Implementado |
27 | 5 | ■ Crear/editar/eliminar cursos | ■ Implementado |
28 | 6 | ■ Crear/editar/eliminar asignaturas | ■ Implementado |
29
30 ### Asignaciones Docentes
31 | # | Funcionalidad | Estado |
32 |---|-----|-----|
33 | 7 | Asignar docente-materia a curso | | Primero seleccionar el curso y de ahi de acuerdo a las asignaturas asignar lo.
34 | 8 | ■ Eliminar asignación docente-materia | ■ Implementado |
35 | 9 | Asignar docente tutor a curso | ■ Implementado |
36 | 10 | Quitar docente tutor de curso | ■ Implementado |
37
38 ### Matriculación
39 | # | Funcionalidad | Estado |
40 |---|-----|-----|
41 | 11 | Matricular estudiantes a cursos | ■ Implementado | mejorar poder matricular varios estudiantes con algun buscador
42 | 12 | Retirar estudiante de curso | ■ Implementado |
43
44 ### Horarios
45 | # | Funcionalidad | Estado |
46 |---|-----|-----|
47 | 13 | Crear horarios de clases por curso | ■ Implementado | Mejorar mas visual
48 | 14 | Editar horarios de clases | | No se si está
49 | 15 | Eliminar clases del horario | ■ Implementado |
50
51 ### Configuración Institucional
52 | # | Funcionalidad | Estado |
53 |---|-----|-----|
54 | 16 | Configurar datos de institución | ■ Implementado |
55 | 17 | Subir/cambiar logo institucional | ■ Implementado |
56 | 18 | Asignar jornadas a institución | ■ Implementado |
57 | 19 | Quitar jornadas de institución | ■ Implementado |
58
```

```

59 ### Asistencia
60 | # | Funcionalidad | Estado |
61 |---|-----|-----|
62 | 20 | Ver asistencias de todos los cursos | ■ Implementado | Mejorar se debe ver todos los cursos
63 | 21 | Registrar asistencia (cualquier curso) | ■ Implementado | Cambiar de rol y verificar
64 | 22 | Editar asistencia (dentro de 48h) | ■ Implementado |
65 | 31 | Ver calendario de asistencias | ■ Implementado |
66
67 ### Justificaciones
68 | # | Funcionalidad | Estado |
69 |---|-----|-----|
70 | 23 | Ver todas las justificaciones | ■ Implementado | Revisar todo esto
71 | 24 | Aprobar justificaciones | ■ Implementado |
72 | 25 | Rechazar justificaciones | ■ Implementado |
73
74 ### Reportes y Estadísticas
75 | # | Funcionalidad | Estado |
76 |---|-----|-----|
77 | 26 | Generar reportes PDF de asistencia | ■ Implementado |
78 | 27 | Generar reportes Excel de asistencia | ■ Implementado |
79 | 28 | Ver estadísticas generales del sistema | ■ Implementado |
80 | 29 | Ver estadísticas por curso | ■ Implementado |
81 | 30 | Ver top estudiantes con más ausencias | ■ Implementado |
82
83 ### RespalDOS y Logs
84 | # | Funcionalidad | Estado |
85 |---|-----|-----|
86 | 32 | Crear respaldos de base de datos | ■ Implementado |
87 | 33 | Descargar respaldos | ■ Implementado |
88 | 34 | Ver logs de actividad del sistema | |
89
90 ### Representantes
91 | # | Funcionalidad | Estado |
92 |---|-----|-----|
93 | 35 | Vincular representante-estudiante | ■ Implementado | Mejorar no se entiende
94 | 36 | Desvincular representante-estudiante | ■ Implementado |
95 | 37 | Marcar representante como principal | ■ Implementado | pero no hay como cambiar a secundario
96
97 ### Sistema
98 | # | Funcionalidad | Estado |
99 |---|-----|-----|
100 | 38 | Buscar en el sistema (global) | |
101 | 39 | Editar su propio perfil | ■ Implementado |
102 | 40 | Cambiar su propia contraseña | ■ Implementado |
103
104 **Total Autoridad: 40/40 ■**
105
106 ---
107
108 ## ■ INSPECTOR
109
110 ### Visualización de Asistencia
111 | # | Funcionalidad | Estado |
112 |---|-----|-----|
113 | 1 | Ver asistencias de todos los cursos | ■ Implementado | No funciona
114 | 2 | Ver asistencias por fecha | ■ Implementado | Separar e implementar
115 | 3 | Ver asistencias por curso | ■ Implementado | Separar e implementar
116 | 4 | Ver calendario de asistencias | ■ Implementado |
117

```

```

118 ### Justificaciones
119 | # | Funcionalidad | Estado |
120 |---|-----|-----|
121 | 5 | Ver todas las justificaciones pendientes | ■ Implementado |
122 | 6 | Ver justificaciones aprobadas | ■ Implementado |
123 | 7 | Ver justificaciones rechazadas | ■ Implementado |
124 | 8 | Aprobar justificaciones | ■ Implementado |
125 | 9 | Rechazar justificaciones | ■ Implementado |
126 | 10 | Descargar documentos de justificación | ■ Implementado |
127
128 ### Estadísticas y Búsqueda
129 | # | Funcionalidad | Estado |
130 |---|-----|-----|
131 | 11 | Ver estadísticas de asistencia general | |
132 | 12 | Ver estadísticas por curso | |
133 | 13 | Ver top estudiantes con ausencias | |
134 | 14 | Buscar asistencias | |
135 | 15 | Buscar estudiantes | |
136 | 16 | Ver horarios de cursos | |
137
138 ### Perfil
139 | # | Funcionalidad | Estado |
140 |---|-----|-----|
141 | 17 | Editar su propio perfil | ■ Implementado |
142 | 18 | Cambiar su propia contraseña | ■ Implementado |
143
144 ### Casos Especiales
145 | # | Funcionalidad | Estado |
146 |---|-----|-----|
147 | 19 | Registrar asistencia (si dicta clase) | ■■ Depende (si tiene asignación docente) |
148
149 **Total Inspector: 18/19 ■ (1 condicional)**
150
151 ---
152
153 ## ■■■ DOCENTE TUTOR
154 *Tiene rol docente + es tutor de un curso*
155
156 ### Registro de Asistencia
157 | # | Funcionalidad | Estado |
158 |---|-----|-----|
159 | 1 | Ver sus clases programadas del día | |
160 | 2 | Registrar asistencia en sus clases | ■ Implementado | error solo debe ser en sus clases
161 | 3 | Editar asistencia de sus clases (48h) | ■ Implementado | No se retorna la asistencia marcada
162 | 4 | Ver asistencias que registró | ■ Implementado | Error se ven todas, solo deberia ver las de el
163
164 ### Como Tutor del Curso
165 | # | Funcionalidad | Estado |
166 |---|-----|-----|
167 | 5 | Ver asistencias de su curso (como tutor) | |
168 | 6 | Ver calendario de su curso | |
169 | 7 | Ver horario de su curso | |
170 | 8 | Ver lista de estudiantes de su curso | |
171 | 9 | Ver estadísticas de su curso | |
172 | 13 | Ver representantes de sus estudiantes | |
173
174 ### Justificaciones
175 | # | Funcionalidad | Estado |
176 |---|-----|-----|

```



```

177 | 10 | Ver justificaciones de su curso | |
178 | 11 | **Aprobar justificaciones de su curso** | ■ **NO implementado** |
179 | 12 | **Rechazar justificaciones de su curso** | ■ **NO implementado** |
180
181 ### Sistema
182 | # | Funcionalidad | Estado |
183 |---|-----|-----|
184 | 14 | Ver dashboard con métricas de su curso | |
185 | 15 | Buscar estudiantes de su curso | |
186 | 16 | Editar su propio perfil | ■ Implementado |
187 | 17 | Cambiar su propia contraseña | ■ Implementado |
188
189 **Total Docente Tutor: 15/17 ■ (2 pendientes)**
190
191 ---
192
193 ## ■■■■ DOCENTE
194 *Sin ser tutor*
195
196 ### Registro de Asistencia
197 | # | Funcionalidad | Estado |
198 |---|-----|-----|
199 | 1 | Ver sus clases programadas del día | |
200 | 2 | Registrar asistencia en sus clases | |
201 | 3 | Editar asistencia de sus clases (48h) | |
202 | 4 | Ver asistencias que registró | |
203
204 ### Visualización
205 | # | Funcionalidad | Estado |
206 |---|-----|-----|
207 | 5 | Ver asistencias de cursos donde dicta | |
208 | 6 | Ver calendario de sus cursos | |
209 | 7 | Ver horario de sus clases | |
210 | 8 | Ver estudiantes de sus cursos | |
211
212 ### Estadísticas y Sistema
213 | # | Funcionalidad | Estado |
214 |---|-----|-----|
215 | 9 | Ver estadísticas de sus cursos | |
216 | 10 | Ver dashboard con sus métricas | |
217 | 11 | Buscar estudiantes de sus cursos | |
218 | 12 | Editar su propio perfil | ■ Implementado |
219 | 13 | Cambiar su propia contraseña | ■ Implementado |
220
221 **Total Docente: 13/13 ■**
222
223 ---
224
225 ## ■ ESTUDIANTE
226
227 ### Visualización de Asistencia
228 | # | Funcionalidad | Estado |
229 |---|-----|-----|
230 | 1 | Ver su propia asistencia | |
231 | 2 | Ver calendario de su asistencia | |
232 | 3 | Ver estadísticas de su asistencia | |
233
234 ### Información Académica
235 | # | Funcionalidad | Estado |

```

```

236 |---|-----|-----|
237 | 4 | Ver su curso y paralelo | |
238 | 5 | Ver horario de clases | |
239 | 6 | Ver sus docentes | |
240 | 7 | Ver quiénes son sus representantes | |
241
242 ### Justificaciones
243 | # | Funcionalidad | Estado |
244 |---|-----|-----|
245 | 8 | **Justificar sus ausencias** | ■ **Implementado** ■■ **REVISAR NECESIDAD** |
246 | 9 | **Subir documento de justificación** | ■ **Implementado** ■■ **REVISAR NECESIDAD** |
247 | 10 | Ver estado de justificaciones enviadas | ■ Implementado |
248 | 11 | Ver justificaciones aprobadas/rechazadas | ■ Implementado |
249
250 ### Sistema
251 | # | Funcionalidad | Estado |
252 |---|-----|-----|
253 | 12 | Ver dashboard con sus métricas | |
254 | 13 | Editar su propio perfil | ■ Implementado |
255 | 14 | Cambiar su propia contraseña | ■ Implementado |
256
257 **Total Estudiante: 14/14 ■ (2 a revisar)**
258
259 ---
260
261 ## ■■■■■ REPRESENTANTE
262
263 ### Gestión de Representados
264 | # | Funcionalidad | Estado |
265 |---|-----|-----|
266 | 1 | Ver lista de sus representados | |
267 | 2 | Ver asistencia de cada representado | |
268 | 3 | Ver calendario de cada representado | |
269 | 4 | Ver estadísticas de cada representado | |
270
271 ### Información Académica
272 | # | Funcionalidad | Estado |
273 |---|-----|-----|
274 | 5 | Ver curso de cada representado | |
275 | 6 | Ver horario de cada representado | |
276 | 7 | Ver docentes de cada representado | |
277
278 ### Justificaciones
279 | # | Funcionalidad | Estado |
280 |---|-----|-----|
281 | 8 | Justificar ausencias de representados | |
282 | 9 | Subir documentos de justificación | |
283 | 10 | Ver justificaciones enviadas | |
284 | 11 | Ver estado de justificaciones | |
285
286 ### Sistema
287 | # | Funcionalidad | Estado |
288 |---|-----|-----|
289 | 12 | Ver dashboard con métricas de representados | |
290 | 13 | Editar su propio perfil | ■ Implementado |
291 | 14 | Cambiar su propia contraseña | ■ Implementado |
292
293 **Total Representante: 14/14 ■**
294

```

```

295 ---
296
297 ## ■ RESUMEN GENERAL
298
299 | Rol | Total Funcionalidades | ■ Implementadas | ■ Faltantes | ■■ A Revisar |
300 |-----|-----|-----|-----|-----|
301 | **Autoridad** | 40 | 40 | 0 | 0 |
302 | **Inspector** | 19 | 18 | 0 | 1 |
303 | **Docente Tutor** | 17 | 15 | 2 | 0 |
304 | **Docente** | 13 | 13 | 0 | 0 |
305 | **Estudiante** | 14 | 14 | 0 | 2 |
306 | **Representante** | 14 | 14 | 0 | 0 |
307 | **TOTAL** | **117** | **114** | **2** | **3** |
308
309 ### Cobertura: **97.4%**
310
311 ---
312
313 ## ■■ PUNTOS A REVISAR/IMPLEMENTAR
314
315 ### ■ FALTANTES (2)
316
317 ##### 1. Docente Tutor - Aprobar justificaciones de su curso
318 - **Estado:** NO implementado
319 - **Pregunta:** ¿Debe implementarse?
320 - **Impacto:** Permitiría descentralizar aprobación de justificaciones
321 - **Acción sugerida:**
322   - Si SÍ → Crear lógica en `JustificationController`
323   - Agregar permiso especial para tutores
324   - Limitar solo a justificaciones de SU curso
325
326 ##### 2. Docente Tutor - Rechazar justificaciones de su curso
327 - **Estado:** NO implementado
328 - **Pregunta:** ¿Debe implementarse?
329 - **Impacto:** Complementa la funcionalidad #1
330 - **Acción sugerida:**
331   - Si SÍ → Agregar al mismo controller
332   - Mismo permiso que aprobación
333
334 ---
335
336 ### ■■ A REVISAR (3)
337
338 ##### 1. Estudiante - Justificar sus propias ausencias
339 - **Estado:** actualmente
340 - **Problema detectado:**
341   - Podría prestarse para abusos
342   - Estudiantes podrían justificar cualquier ausencia
343   - Dificulta control institucional
344 - **Ventaja:**
345   - Ayuda cuando padre no tiene acceso a computadora
346   - Estudiantes responsables pueden gestionar
347 - **Opciones:**
348   1. **Mantener** (con advertencia de uso responsable)
349   2. **Deshabilitar** (solo representante justifica)
350   3. **Requiere aprobación adicional** (estudiante envía → representante confirma → autoridad aprueba)
351   4. **Limitar cantidad** (máximo X justificaciones por mes)
352
353 ##### 2. Estudiante - Subir documentos de justificación

```

```
354 - **Estado:** actualmente
355 - **Problema:** Mismo que punto #1
356 - **Acción:** Depende de decisión del punto anterior
357
358 ##### 3. Inspector - Registrar asistencia si dicta clase
359 - **Estado:** ■■ Depende
360 - **Situación actual:**
361   - SI tiene asignación como docente → Sí puede registrar
362   - NO tiene asignación → NO puede registrar
363 - **Lógica:** Correcta según rol híbrido
364 - **Acción:** Mantener como está
365
366 ---
367
368 ## ■ RECOMENDACIONES
369
370 ### Prioridad ALTA
371 1. **Decidir sobre justificaciones de estudiantes**
372   - Analizar casos de uso reales
373   - Consultar con autoridades educativas
374   - Implementar solución definitiva
375
376 2. **Implementar aprobación de justificaciones por tutor**
377   - Descentraliza carga de trabajo
378   - Tutor conoce mejor a sus estudiantes
379   - Acelera proceso de aprobación
380
381 ### Prioridad MEDIA
382 3. **Agregar logs de justificaciones**
383   - Quién justificó (estudiante/representante)
384   - Quién aprobó (autoridad/inspector/tutor)
385   - Auditoría completa
386
387 4. **Notificaciones automáticas**
388   - Cuando se justifica → notificar tutor/autoridad
389   - Cuando se aprueba → notificar estudiante/representante
390
391 ### Prioridad BAJA
392 5. **Estadísticas de justificaciones**
393   - Estudiantes con más justificaciones
394   - Porcentaje de aprobación
395   - Motivos más frecuentes
396
397 ---
398
399 ## ■ NOTAS FINALES
400
401 - Sistema altamente funcional con **97.4% de cobertura**
402 - Faltantes son **decisiones de diseño**, no errores técnicos
403 - Arquitectura permite agregar funcionalidades fácilmente
404 - Código preparado para escalar
405
406 ---
407
408 **Documento generado:** 16 de Febrero de 2026
409 **Sistema:** EcuAsist 2026 v1.5
410 **Autor:** Análisis de funcionalidades por rol
```

3. README.md

```
1  ## EcuAsistencía2026
2
3  # EcuAsist 2026 - SisteMa de Asistencía Escollar
4
5
6
7  SisteMa de gestioN de asistencía Escollar de sarrolad
8
9
10
11  ### = CAracterísticAs
12
13
14
15  ##### Modulos Principales
16
17  - Autenticación y roles multi-usuari
18
19  - GestioN de usuarios (esetuidiantes, docentes, aut
20
21  - Registro de asistencía con validación de feh
22
23  - SisteMa de justificación con aprobac
24
25  - Reportes en PDF y Excel
26
27  - Cálculo de arío de asistencía
28
29  - SisteMa de horari
30
31  - Configuración de insstituc
32
33  - GestioN de jornadas por insstituc
34
35  - GestioN de separada de tut
36
37  - EstadísticAs avanzadas
38
39  - Asignación de docentes y tut
40
41  - Relación de re
42
43  - SisteMa de respaldos
44
45  - Logs de activid
46
47
48
49  ##### Tecnolog
50
51  - PHP 7.4+
52
53  - MySQL 5.7+ / MariaDB
54
55  - PDO para bases de datos
56
57  - PHPMailer para correo
58
```

59 ■-■ ■T■C■P■D■F■ ■p■a■r■a■ ■P■D■F■
60 ■
61 ■-■ ■P■h■p■S■p■r■e■a■d■s■h■e■e■t■ ■p■a■r■a■ ■E■x■c■e■l■
62 ■
63 ■
64 ■
65 ■#■#■ ■=■■■■ ■I■n■s■t■a■l■a■c■i■■■n■
66 ■
67 ■
68 ■
69 ■#■#■#■ ■R■e■q■u■i■s■i■t■o■s■
70 ■
71 ■-■ ■X■A■M■P■P■ ■/■ ■W■A■M■P■ ■/■ ■L■A■M■P■
72 ■
73 ■-■ ■P■H■P■ ■>■=■ ■7■.■4■
74 ■
75 ■-■ ■M■y■S■Q■L■ ■>■=■ ■5■.■7■
76 ■
77 ■-■ ■C■o■m■p■o■s■e■r■
78 ■
79 ■
80 ■
81 ■#■#■#■ ■P■a■s■o■s■
82 ■
83 ■
84 ■
85 ■1■.■ ■*■*■C■l■o■n■a■r■ ■r■e■p■o■s■i■t■o■r■i■o■*■*■
86 ■
87 ■`■`■`■b■a■s■h■
88 ■
89 ■g■i■t■ ■c■l■o■n■e■ ■h■t■t■p■s■:■/■/■g■i■t■h■u■b■.■c■o■m■/■t■u■-■u■s■u■a■r■i■o■/■e■c■u■a■s■i■s■t■-■2■0■2■6■
90 ■
91 ■c■d■ ■e■c■u■a■s■i■s■t■-■2■0■2■6■
92 ■
93 ■`■`■`■
94 ■
95 ■
96 ■
97 ■2■.■ ■*■*■I■n■s■t■a■l■a■r■ ■d■e■p■e■n■d■e■n■c■i■a■s■*■*■
98 ■
99 ■`■`■`■b■a■s■h■
100 ■
101 ■c■o■m■p■o■s■e■r■ ■i■n■s■t■a■l■l■
102 ■
103 ■`■`■`■
104 ■
105 ■
106 ■
107 ■3■.■ ■*■*■C■r■e■a■r■ ■b■a■s■e■ ■d■e■ ■d■a■t■o■s■*■*■
108 ■
109 ■-■ ■I■m■p■o■r■t■a■r■ ■`■e■c■u■a■s■i■s■t■e■n■c■i■a■2■0■2■6■_■d■b■.■s■q■l■`■ ■e■n■ ■p■h■p■M■y■A■d■m■i■n■
110 ■
111 ■
112 ■
113 ■4■.■ ■*■*■C■o■n■f■i■g■u■r■a■r■*■*■
114 ■
115 ■-■ ■E■d■i■t■a■r■ ■`■c■o■n■f■i■g■/■d■a■t■a■b■a■s■e■.■p■h■p■`■ ■c■o■n■ ■c■r■e■d■e■n■c■i■a■l■e■s■ ■M■y■S■Q■L■
116 ■
117 ■-■ ■E■d■i■t■a■r■ ■`■c■o■n■f■i■g■/■c■o■n■f■i■g■.■p■h■p■`■ ■c■o■n■ ■U■R■L■ ■b■a■s■e■ ■y■ ■c■o■n■f■i■g■u■r■a■

118 ■
119 ■
120 ■
121 ■6■.■ ■*■*■C■r■e■a■r■ ■c■a■r■p■e■t■a■s■ ■d■e■ ■u■p■l■o■a■d■s■*■*■
122 ■
123 ■`■`■`■b■a■s■h■
124 ■
125 ■m■k■d■i■r■ ■u■p■l■o■a■d■s■/■i■n■s■t■i■t■u■t■i■o■n■
126 ■
127 ■
128 ■
129 ■7■.■ ■*■*■p■e■r■m■i■s■o■s■*■*■ ■(■f■i■n■u■x■/■M■a■c■)■*■*■
130 ■
131 ■`■`■`■b■a■s■h■
132 ■
133 ■c■h■m■o■d■ ■-■R■ ■7■5■5■ ■.■
134 ■
135 ■c■h■m■o■d■ ■-■R■ ■7■7■7■ ■u■p■l■o■a■d■s■
136 ■
137 ■c■h■m■o■d■ ■-■R■ ■7■7■7■ ■b■a■c■k■u■p■s■
138 ■
139 ■`■`■`■`■
140 ■
141 ■
142 ■
143 ■8■.■ ■*■*■A■c■c■e■d■e■r■*■*■
144 ■
145 ■`■`■`■`■
146 ■
147 ■h■t■t■p■:■/■/■l■o■c■a■l■h■o■s■t■/■e■c■u■a■s■i■s■t■e■n■c■i■a■2■0■2■6■/■p■u■b■l■i■c■/■
148 ■
149 ■`■`■`■`■
150 ■
151 ■
152 ■
153 ■#■#■ ■=■d■ ■C■r■e■d■e■n■c■i■a■l■e■s■ ■p■o■r■ ■D■e■f■e■c■t■o■
154 ■
155 ■
156 ■
157 ■U■s■u■a■r■i■o■ ■d■e■ ■p■r■u■e■b■a■ ■(■d■e■s■p■u■s■ ■d■e■ ■i■m■p■o■r■t■a■r■ ■S■Q■L■ ■d■e■ ■p■r■u■e■b■a■)
158 ■
159 ■-■ ■*■*■U■s■u■a■r■i■o■:■*■*■ ■p■r■o■f■.■g■a■r■c■i■a■
160 ■
161 ■-■ ■*■*■C■o■n■t■r■a■s■e■a■:■*■*■ ■p■a■s■s■w■o■r■d■
162 ■
163 ■
164 ■
165 ■#■#■ ■=■ ■E■s■t■r■u■c■t■u■r■a■ ■d■e■l■ ■P■r■o■y■e■c■t■o■
166 ■
167 ■`■`■`■`■
168 ■
169 ■e■c■u■a■s■i■s■t■/■
170 ■
171 ■%■%■%■ c■o■n■f■i■g■/■ ■ ■ ■ ■ ■ ■ ■ ■ ■#■ C■o■n■f■i■g■u■r■a■c■i■n■
172 ■
173 ■%■%■%■ c■o■n■t■r■o■l■l■e■r■s■/■ ■ ■ ■ ■ ■ ■ ■ ■ ■#■ C■o■n■t■r■o■l■a■d■o■r■e■s■
174 ■
175 ■%■%■%■ m■o■d■e■l■s■/■ ■ ■ ■ ■ ■ ■ ■ ■ ■#■ M■o■d■e■l■o■s■ ■d■e■ ■d■a■t■o■s■
176 ■

[illegible]

236 ■
237 ■-■ ■F■i■n■e■s■ ■d■e■ ■s■e■m■a■n■a■ ■n■o■ ■c■u■e■n■t■a■n■ ■e■n■ ■e■l■ ■c■o■l■l■c■u■l■o■ ■d■e■ ■4■8■ ■h■o■r■a■
238 ■
239 ■-■ ■E■d■i■c■i■o■n■ ■d■e■ ■a■s■i■s■t■e■n■c■i■a■:■ ■h■a■s■t■a■ ■4■8■ ■h■o■r■a■s■ ■d■e■s■p■u■s■
240 ■
241 ■
242 ■
243 ■#■#■ ■=■■■■■■ ■S■e■g■u■r■i■d■a■d■
244 ■
245 ■
246 ■
247 ■-■ ■C■o■n■t■r■a■s■e■a■s■ ■h■a■s■h■e■a■d■a■s■ ■c■o■n■ ■`p■a■s■s■w■o■r■d■_h■a■s■h■(■)■`■
248 ■
249 ■-■ ■P■r■o■t■e■c■c■i■o■n■ ■C■S■R■F■ ■e■n■ ■f■o■r■m■u■l■a■r■i■o■s■
250 ■
251 ■-■ ■V■a■l■i■d■a■c■i■o■n■ ■d■e■ ■e■n■t■r■a■d■a■s■
252 ■
253 ■-■ ■S■a■n■i■t■i■z■a■c■i■o■n■ ■d■e■ ■d■a■t■o■s■
254 ■
255 ■-■ ■C■o■n■t■r■o■l■ ■d■e■ ■a■c■c■e■s■o■ ■p■o■r■ ■r■o■l■l■e■s■
256 ■
257 ■
258 ■
259 ■#■#■ ■=■■■■ ■S■o■p■o■r■t■e■
260 ■
261 ■
262 ■
263 ■P■a■r■a■ ■s■o■p■o■r■t■e■ ■o■ ■c■o■n■s■u■l■t■a■s■:■ ■s■o■p■o■r■t■e■@■e■c■u■a■s■i■s■t■,■e■d■u■,■e■c■
264 ■
265 ■
266 ■
267 ■#■#■ ■=■■■■ ■L■i■c■e■n■c■i■a■
268 ■
269 ■
270 ■
271 ■P■r■o■y■e■c■t■o■ ■e■d■u■c■a■t■i■v■o■ ■-■ ■U■s■o■ ■l■i■b■r■e■ ■c■o■n■ ■a■t■r■i■b■u■c■i■o■n■
272 ■
273 ■
274 ■
275 ■#■#■ ■=■'■ ■M■a■n■t■e■n■i■m■i■e■n■t■o■
276 ■
277 ■
278 ■
279 ■#■#■#■ ■R■e■s■p■a■l■d■o■s■
280 ■
281 ■-■ ■A■u■t■o■m■a■t■i■c■o■s■ ■e■n■ ■`■/■b■a■c■k■u■p■s■`■
282 ■
283 ■-■ ■S■e■ ■r■e■c■o■m■i■e■n■d■a■ ■d■e■s■c■a■r■g■a■r■ ■s■e■m■a■n■a■l■m■e■n■t■e■
284 ■
285 ■-■ ■L■i■m■p■i■e■z■a■ ■a■u■t■o■m■a■t■i■c■a■ ■d■e■ ■>300 ■d■a■s■
286 ■
287 ■
288 ■
289 ■#■#■#■ ■L■o■g■s■
290 ■
291 ■-■ ■A■c■t■i■v■i■d■a■d■ ■d■e■ ■u■s■u■a■r■i■o■s■ ■e■n■ ■`a■c■t■i■v■i■t■y■_l■o■g■s■`■
292 ■
293 ■-■ ■R■e■v■i■s■a■r■ ■p■e■r■i■d■i■c■a■m■e■n■t■e■ ■e■n■ ■p■a■n■e■l■ ■d■e■ ■a■u■t■o■r■i■d■a■d■
294 ■

295 ■
296 ■
297 ■*■*■E■n■ R■o■a■d■m■a■p■ ■(■r■e■e■m■p■l■a■z■a■r■ t■o■d■o■)■:■*■*■
298 ■
299 ■`■`■`■m■a■r■k■d■o■w■n■
300 ■
301 ■-■ ■{■x■}■ S■i■s■t■e■m■a■ d■e■ h■o■r■a■r■i■o■s■
302 ■
303 ■-■ ■{■x■}■ C■o■n■f■i■g■u■r■a■c■i■o■n■ m■u■l■t■i■-■j■o■r■n■a■d■a■
304 ■
305 ■-■ ■{■x■}■ P■r■o■v■i■n■c■i■a■s■/■c■i■u■d■a■d■e■s■ E■c■u■a■d■o■r■
306 ■
307 ■-■ ■{■ ■}■ G■r■f■i■c■o■s■ i■n■t■e■r■a■c■t■i■v■o■s■
308 ■
309 ■-■ ■{■ ■}■ B■s■q■u■e■d■a■ g■l■o■b■a■l■
310 ■
311 ■-■ ■{■ ■}■ M■o■d■o■ o■s■c■u■r■o■
312 ■
313 ■
314 ■
315 ■-■-■-■
316 ■
317 ■D■e■s■a■r■r■o■l■l■a■d■o■ c■o■n■ d'■ p■a■r■a■ i■n■s■t■i■t■u■c■i■o■n■e■s■ e■d■u■c■a■t■i■v■a■s■ e■c■u■

4. ResumenEjecutivo_08C.md

1 ■ ECUASISTENCIA 2026
2 Sistema Integral de Gestión de Asistencia Escolar
3
4 Versión consolidada: v2.0
5 Fecha: Febrero 2026
6 Estado: Sistema completo - Optimización y validación productiva
7
8 ■ ESTADO GENERAL DEL PROYECTO
9
10 Sistema desarrollado en PHP 7.4+ OOP puro + MySQL, arquitectura MVC modular, sin frameworks externos.
11
12 ■ 19/19 módulos completados
13 ■ 0 bugs críticos conocidos
14 ■ Diseño visual unificado
15 ■ Seguridad reforzada
16 ■ Superusuario global implementado
17 ■ Fase actual: optimización avanzada y pruebas productivas
18
19 El sistema ya no está en construcción base. Está en etapa de refinamiento y validación real.
20
21 ■■ ARQUITECTURA DEL SISTEMA
22 Backend
23
24 Arquitectura MVC modular
25
26 Router central: public/index.php
27
28 Controllers independientes por módulo
29
30 Models especializados
31
32 Helpers: Seguridad, Correo, Respaldo
33
34 BASE_PATH dinámico
35
36 Detección automática entorno local / producción
37
38 Control de sesiones 24h
39
40 Timeout automático 30 min
41
42 ■■ BASE DE DATOS
43
44 Base normalizada con restricciones y validaciones activas.
45
46 Principales
47
48 institutions
49
50 users
51
52 roles
53
54 permissions
55
56 Académico
57
58 school_years

59
60 courses
61
62 subjects
63
64 teacher_assignments
65
66 course_students
67
68 class_schedule
69
70 institution_shifts
71
72 course_subjects (tabla nueva con UNIQUE y CASCADE)
73
74 Asistencia
75
76 attendances
77
78 justifications
79
80 Sistema
81
82 notifications
83
84 activity_logs
85
86 representatives
87
88 Validaciones activas:
89
90 Tutor único por curso
91
92 Materia única por curso
93
94 Horarios sin conflictos
95
96 Asistencia sin duplicados
97
98 Estudiante único por año lectivo
99
100 Jornadas controladas
101
102 ■ SISTEMA DE DISEÑO UNIFICADO (v2.0)
103
104 Se eliminó completamente Bootstrap y CSS por vista.
105 Ahora existe un sistema visual centralizado.
106
107 Archivo central:
108
109 public/global.css
110
111 Componentes:
112
113 .page-header
114
115 .panel
116
117 Botones estandarizados

118
119 Formularios unificados
120
121 Breadcrumbs globales
122
123 Empty states
124
125 Tablas con hover
126
127 Variables CSS del sistema
128
129 Resultado:
130 Interfaz limpia, consistente y sin conflictos.
131
132 Container unificado a 1200px.
133
134 ■ NAVBAR PROFESIONAL
135
136 Sticky
137
138 Responsive total
139
140 Menú hamburguesa móvil
141
142 Dropdowns organizados por módulo
143
144 Indicador activo
145
146 Tooltip con Apellido, Nombre
147
148 Avatar con iniciales
149
150 Notificaciones en tiempo real
151
152 Acceso dinámico por rol
153
154 Logo institucional dinámico con img.php
155
156 Fondo profesional oscuro
157
158 ■ SUPERUSUARIO GLOBAL
159
160 Nueva columna is_superuser
161
162 Bypass total de permisos
163
164 Acceso completo a módulos
165
166 Flag en sesión
167
168 Script SQL de reseteo
169
170 Admin ID=1 como superusuario absoluto
171
172 Seguridad reforzada.
173
174 ■ CONFIGURACIÓN INSTITUCIONAL AVANZADA
175
176 Campos ampliados:

177
178 Provincia
179
180 Ciudad
181
182 Director
183
184 Código AMIE
185
186 Web
187
188 Logo institucional
189
190 Jornadas
191
192 Mejoras técnicas:
193
194 img.php en raíz (seguro contra path traversal)
195
196 Triple respaldo logo_path
197
198 Cache-busting automático
199
200 Permisos 775 para hosting compartido
201
202 BASE_URL y BASE_PATH automáticos
203
204 ■ AÑOS LECTIVOS
205
206 Corrección crítica:
207
208 lastInsertId() usa misma conexión
209
210 Funciones:
211
212 Activación inmediata
213
214 Modal personalizado
215
216 Gestión estable
217
218 ■■■ SISTEMA DE HORARIOS
219
220 Tabla class_schedule
221
222 Validación anti-duplicados
223
224 Detección de conflictos en tiempo real
225
226 Auto-asignación docente
227
228 Flujo automatizado
229
230 ■ REGISTRO DE ASISTENCIA INTELIGENTE
231
232 Detecta clases del día automáticamente
233
234 Sin selección manual de curso
235

236 Validación 48h hábiles
237
238 Actualiza si existe registro
239
240 Precarga estados guardados
241
242 Sin duplicados
243
244 ■ SISTEMA DE JUSTIFICACIONES AVANZADO
245
246 Nuevas columnas:
247
248 date_from
249
250 date_to
251
252 working_days
253
254 reason_type
255
256 can_approve
257
258 attendance_id nullable
259
260 Lógica:
261
262 ≤ 3 días → Tutor
263
264 3 días → Inspector / Autoridad
265
266 Solo muestra ausencias reales
267
268 Contador dinámico
269
270 9 causas predefinidas
271
272 Validaciones completas antes de enviar
273
274 Sistema sólido y coherente con normativa real.
275
276 ■■■ MÓDULO DOCENTE TUTOR
277
278 Dashboard avanzado:
279
280 Métricas completas
281
282 Barra de efectividad
283
284 Estadísticas del día
285
286 Últimos 7 días
287
288 Top 10 ausencias
289
290 Búsqueda avanzada
291
292 Colores por porcentaje:
293
294 Verde $\geq 90\%$

295
296 Amarillo $\geq 75\%$
297
298 Rojo $\leq 75\%$
299
300 ■ GESTIÓN DE USUARIOS
301
302 Validación cédula ecuatoriana (algoritmo módulo 10)
303
304 Validación teléfono Ecuador
305
306 Toggle extranjero / pasaporte
307
308 Filtro por rol con botones tipo píldora
309
310 Seguridad backend y frontend
311
312 ■■ MÓDULO DE RESPALDOS
313
314 Detección automática mysqldump
315
316 Validación de archivos
317
318 Eliminación individual
319
320 Limpieza automática
321
322 Pendiente menor: ruta delete_backup
323
324 ■ REPORTE
325
326 PDF con TCPDF
327
328 Excel con PhpSpreadsheet
329
330 Datos institucionales dinámicos
331
332 Entidades HTML corregidas
333
334 Nombres sanitizados
335
336 Sin duplicidades
337
338 ■ BUGS CRÍTICOS CORREGIDOS
339
340 lastInsertId incorrecto
341
342 Logo no actualizaba
343
344 CSS conflictivos
345
346 Rutas faltantes
347
348 Backups vacíos
349
350 Justificaciones con fechas inválidas
351
352 Entidades mal renderizadas
353

354 Vista accediendo a modelo directamente
355
356 Estado actual: 0 bugs críticos conocidos.
357
358 ■■ CONFIGURACIÓN PRODUCCIÓN
359
360 Requisitos clave:
361
362 img.php en raíz
363
364 uploads/ con permisos 775
365
366 env.php configurado correctamente
367
368 Tabla course_subjects creada
369
370 SMTP activo
371
372 Contraseña admin cambiada
373
374 Linux:
375
376 chmod -R 755 .
377 chmod -R 775 uploads backups
378
379 Zona horaria Ecuador activa.
380
381 ■ DEPENDENCIAS
382
383 phpmailer/phpmailer
384
385 phppoffice/phpspreadsheet
386
387 tecnickcom/tcpdf
388
389 ■ ESTADO GLOBAL
390 Área■Estado
391 Arquitectura■Estable
392 Módulos■Completos
393 Seguridad■Sólida
394 UX/UI■Unificada
395 Responsive■Implementado
396 Base de datos■Normalizada
397 Optimización■En progreso
398 ■ PRÓXIMOS PASOS ESTRATÉGICOS
399
400 Alta prioridad:
401
402 Validación completa con datos reales
403
404 Pruebas de rendimiento en estadísticas
405
406 Optimización consultas pesadas
407
408 Media:
409
410 Toast notifications
411
412 Gráficos avanzados

413
414 Breadcrumbs finales
415
416 Baja:
417
418 Modo oscuro
419
420 Calendario visual
421
422 API REST
423
424 Analítica avanzada
425
426 Mensajería interna
427
428 ■ CONCLUSIÓN FINAL
429
430 EcuAsistencia 2026 v2.0 alcanzó madurez funcional completa:
431
432 ✓ Automatización académica sólida
433 ✓ Justificaciones inteligentes por días reales
434 ✓ Dashboard tutor avanzado
435 ✓ Seguridad reforzada con superusuario
436 ✓ Arquitectura modular escalable
437 ✓ Diseño visual profesional unificado
438
439 El sistema está listo para validación institucional controlada y despliegue productivo.

5. SugerenciasFuturas.md

```
1  ■ SUGERENCIAS PARA MEJORAR LA INTUICIÓN DEL SISTEMA
2  ■ 1. NAVEGACIÓN Y MENÚ
3  Actual:
4
5  Menú con texto técnico
6  Dropdown con múltiples opciones
7
8  Mejoras:
9  ■ Iconos grandes y visuales en lugar de solo texto
10 ■ Breadcrumbs (Inicio &gt; Asistencia &gt; Registrar) para saber dónde está
11 ■ Menú contextual según el rol (que solo vea lo que puede hacer)
12 ■ Destacar acciones principales con colores llamativos
13 ■ Búsqueda global en el navbar para encontrar cualquier cosa rápido
14
15 ■ 2. DASHBOARD
16 Actual:
17
18 Lista de enlaces
19 Sin guía visual
20
21 Mejoras:
22 ■ Tarjetas grandes con iconos tipo "Apps" del móvil
23 ■ Acciones rápidas destacadas ("Registrar Asistencia HOY")
24 ■ Tutorial interactivo la primera vez que entran
25 ■ Atajos del teclado (Ctrl+R = Registrar asistencia)
26 ■ Widget de "Últimas acciones" para continuar donde dejó
27
28 ■ 3. REGISTRO DE ASISTENCIA
29 Actual:
30
31 Seleccionar clase de lista
32 Varios pasos
33
34 Mejoras:
35 ■ Auto-detectar clase actual según hora del día
36 ■ Vista tipo calendario con clases del día
37 ■ Marcar todos como "Presente" por defecto con un click
38 ■ Iconos visuales (■ ■ ■) en lugar de dropdown
39 ■ Atajos de teclado (P=Presente, A=Ausente, T=Tardanza)
40 ■ Confirmación visual con animación al guardar
41 ■ Fotos de estudiantes (opcional) para identificar rápido
42
43 ■■ 4. HORARIOS
44 Actual:
45
46 Formulario con dropdowns
47 Vista de tabla
48
49 Mejoras:
50 ■ Vista de parrilla visual (como Google Calendar)
51 ■ Drag & drop para mover clases
52 ■ Colores por asignatura para identificar rápido
53 ■ Copiar horario de semana anterior con un click
54 ■ Plantillas predefinidas de horarios comunes
55
56 ■ 5. GESTIÓN DE USUARIOS
57 Actual:
58
```

59 Tabla con filtros
60 Dropdowns para asignar roles
61
62 Mejoras:
63 ■ Tarjetas con foto en lugar de tabla
64 ■ Vista tipo "contactos" del móvil
65 ■ Búsqueda en tiempo real mientras escribes
66 ■ Etiquetas de colores para roles
67 ■ Acciones rápidas con iconos (■■ ver, =■ editar, ■■ eliminar)
68
69 ■ 6. REPORTES
70 Actual:
71
72 Formulario con fechas
73 Vista previa en tabla
74
75 Mejoras:
76 ■ Plantillas de reportes ("Reporte semanal", "Mensual", "Por estudiante")
77 ■ Gráficos visuales en lugar de solo tablas
78 ■ Comparativas automáticas (este mes vs anterior)
79 ■ Exportar con un click (PDF/Excel visible siempre)
80 ■ Previsualizaciones en miniatura antes de descargar
81
82 ■ 7. NOTIFICACIONES Y FEEDBACK
83 Actual:
84
85 Mensajes de texto simples
86 Popups del navegador
87
88 Mejoras:
89 ■ Toast notifications con iconos y colores
90 ■ Sonidos sutiles al completar acciones
91 ■ Progreso visual en operaciones largas
92 ■ Deshacer acción después de guardar (5 segundos)
93 ■ Centro de notificaciones con campana en navbar
94
95 ■ 8. AYUDA CONTEXTUAL
96 Actual:
97
98 Sin ayuda
99 Usuario debe adivinar
100
101 Mejoras:
102 ■ Tooltips en hover sobre campos
103 ■ Videos tutoriales cortos (15-30 seg) en cada módulo
104 ■ Asistente virtual tipo chatbot
105 ■ Modo guiado que explica paso a paso
106 ■ FAQs integradas en cada pantalla
107 ■ Botón de "¿Necesitas ayuda?" siempre visible
108
109 ■ 9. FLUJOS SIMPLIFICADOS
110 Mejoras específicas:
111 Asignar Tutor:
112 ■ Antes: Curso → Docente (con validaciones)
113 ■ Ahora: Ver todos los cursos → Click en "Asignar tutor" → Elegir docente disponible
114 Crear Horario:
115 ■ Antes: Formulario → Día → Hora → Materia → Docente
116 ■ Ahora: Arrastrar asignatura al calendario visual
117 Justificar Ausencia:

118 ■ Antes: Buscar ausencia → Click justificar → Formulario
119 ■ Ahora: Notificación "Tienes 2 ausencias sin justificar" → Click → Justificar en popup
120
121 ■ 10. ELEMENTOS VISUALES
122 Mejoras:
123 ■ Colores semánticos consistentes:
124
125 ■ Verde = Éxito/Presente
126 ■ Rojo = Error/Ausente
127 ■ Amarillo = Advertencia/Tardanza
128 ■ Azul = Información/Acciones
129
130 ■ Estados visuales claros:
131
132 Botones deshabilitados en gris claro
133 Campos requeridos con asterisco rojo (*)
134 Campos opcionales con "(opcional)"
135
136 ■ Animaciones sutiles:
137
138 Fade in/out en modales
139 Slide para notificaciones
140 Efecto de "check" al guardar
141
142
143 ■ 11. RESPONSIVE Y MÓVIL
144 Mejoras:
145 ■ Botones grandes para touch
146 ■ Menú hamburguesa en móvil
147 ■ Swipe gestures (deslizar para eliminar)
148 ■ Vista adaptada (tarjetas en móvil, tabla en escritorio)
149
150 ■ 12. ATAJOS Y PRODUCTIVIDAD
151 Mejoras:
152 ■ Comandos rápidos (Ctrl+K = Buscar)
153 ■ Historial de acciones para repetir rápido
154 ■ Acciones masivas (Seleccionar varios → Acción)
155 ■ Auto-completar en campos de texto
156 ■ Recordar preferencias (último curso seleccionado)
157
158 ■ 13. PERSONALIZACIÓN
159 Mejoras:
160 ■ Modo oscuro/claro
161 ■ Tamaño de fuente ajustable
162 ■ Dashboard personalizable (mover widgets)
163 ■ Favoritos para accesos directos
164
165 ■ 14. PREVENCIÓN DE ERRORES
166 Mejoras:
167 ■ Validación en tiempo real (mensaje aparece al escribir)
168 ■ Autocompletado inteligente
169 ■ Confirmación de acciones críticas con modal visual
170 ■ Guardar automático cada X segundos
171 ■ Indicadores de estado (guardando..., guardado ✓)
172
173 ■ PRIORIDAD DE IMPLEMENTACIÓN
174 Alta (Hacer YA):
175
176 Iconos grandes en dashboard

177 Breadcrumbs
178 Tooltips en campos
179 Modales mejorados (ya hechos ✓)
180 Notificaciones toast
181 Auto-detectar clase actual
182
183 Media (Hacer pronto):
184
185 Gráficos visuales
186 Búsqueda global
187 Historial de acciones
188 Modo oscuro
189 Vista calendario para horarios
190 Fotos de estudiantes
191
192 Baja (Mejoras futuras):
193
194 Asistente virtual
195 Videos tutoriales
196 Drag & drop
197 Atajos de teclado
198 Personalización avanzada
199
200
201 ¿Empezamos con las de prioridad ALTA?

6. malla.md

```
1 ■ Educación Inicial
2 Inicial 1-2
3 •■Desarrollo personal y social
4 •■Expresión y comunicación
5 •■Relación con el entorno natural y cultural
6
7 ■ Educación General Básica (1.º a 10.º)
8 Subniveles Preparatoria → Superior
9 • Lengua y Literatura
10 • Matemática
11 • Entorno Natural y Social (primeros niveles)
12 • Estudios Sociales
13 • Ciencias Naturales
14 • Educación Física
15 • Inglés
16 • educación financiera
17 • socioemocional
18 • cívica
19 • sostenibilidad
20 • seguridad vial
21 ■ Bachillerato General Unificado (BGU)
22 1.º BGU
23 •■Matemática
24 •■Física
25 •■Química
26 •■Biología
27 •■Historia
28 •■Educación para la Ciudadanía
29 •■Filosofía
30 •■Lengua y Literatura
31 •■Inglés
32 •■Educación Cultural y Artística
33 •■Educación Física
34 •■Emprendimiento y Gestión
35 2.º BGU
36 •■Matemática
37 •■Física
38 •■Química
39 •■Biología
40 •■Historia
41 •■Educación para la Ciudadanía
42 •■Filosofía
43 •■Lengua y Literatura
44 •■Inglés
45 •■Educación Cultural y Artística
46 •■Educación Física
47 •■Emprendimiento y Gestión
48 3.º BGU
49 •■Matemática
50 •■Física
51 •■Química
52 •■Biología
53 •■Historia
54 •■Lengua y Literatura
55 •■Inglés
56 •■Educación Física
57 •■Emprendimiento y Gestión
58 Optativas institucionales (ejemplos reales):
```

59	•■Matemática superior
60	•■Física superior
61	•■Biología superior
62	•■Desarrollo del pensamiento abstracto
63	Bachillerato Técnico – Asignaturas por nivel
64	■ 1.º BT
65	•■Lengua y Literatura
66	•■Matemática
67	•■Física
68	•■Química
69	•■Biología
70	•■Historia
71	•■Educación para la Ciudadanía
72	•■Filosofía
73	•■Inglés
74	•■Educación Física
75	•■Educación Cultural y Artística
76	•■Emprendimiento y Gestión
77	•■Módulos técnicos de la figura profesional
78	■ 2.º BT
79	•■Lengua y Literatura
80	•■Matemática
81	•■Física
82	•■Química
83	•■Biología
84	•■Historia
85	•■Educación para la Ciudadanía
86	•■Filosofía
87	•■Inglés
88	•■Educación Física
89	•■Educación Cultural y Artística
90	•■Emprendimiento y Gestión
91	•■Módulos técnicos de la figura profesional
92	■ 3.º BT
93	•■Lengua y Literatura
94	•■Matemática
95	•■Física
96	•■Química
97	•■Biología
98	•■Historia
99	•■Inglés
100	•■Educación Física
101	•■Emprendimiento y Gestión
102	•■Módulos técnicos de la figura profesional
103	Matriz de carga horaria – Sistema educativo Ecuador
104	(Períodos mínimos semanales – referencia curricular vigente)
105	■ Inicial
106	Nivel■Horas semanales
107	Inicial 1■Flexible (no carga rígida por asignatura – currículo por ámbitos)
108	Inicial 2■Flexible (organización institucional)
109	■ No se estructura por materias sino por ámbitos de desarrollo.
110	
111	■ EGB – Preparatoria
112	Grado■Horas
113	1.º EGB■~30 horas semanales
114	•■25 h currículo integrador
115	•■5 h EF + ECA
116	
117	

118 ■ EGB Elemental / Media
119 Grados■Horas
120 2.º - 7.º EGB■mínimo 30 horas semanales
121 Enfocado en comunicación, matemáticas, competencias digitales y socioemocionales.
122
123 _____
124 ■ EGB Superior
125 Grados■Horas
126 8.º - 10.º EGB■~30 horas semanales
127 Distribución típica núcleo:
128 •■Matemática → 6 h
129 •■Lengua → 6 h
130 •■Ciencias Naturales → 4 h
131 •■Estudios Sociales → 4 h
132 •■Otras asignaturas → 1 h c/u
133
134 _____
135 ■ Bachillerato (Ciencias / Técnico)
136 Modalidad■Períodos semanales
137 BGU Ciencias■40 períodos
138 BGU Técnico■45 períodos
139 Carga típica por asignatura núcleo:
140 •■Matemática → 5 h
141 •■Lengua → 5 h
142 •■Física / Química / Biología / Inglés → 3 h
143 •■Historia / Ciudadanía / Filosofía / Arte / Emprendimiento → 2 h
144 •■Cívica → 1 h
145
146 Además:
147 •■Técnico → +10 períodos módulos técnicos
148 •■Ciencias → +5 períodos institucionales
149
150

7. men#U00fa_v2.0.md

```
1 # ■ Menú EcuAsistencia 2026 – v2.0
2
3 ## ■ 1. Dashboard
4 - Resumen institucional del día
5 - Indicadores rápidos (asistencia, alertas, justificativos pendientes)
6 - ■ Notificaciones visuales por rol (badge contador)
7 - ■ Accesos rápidos personalizados por rol
8
9 ## ■ 2. Usuarios y Roles
10 - Gestión de usuarios (crear, editar, desactivar)
11 - Roles y permisos (Autoridad, Inspector, Docente, Estudiante, Representante)
12 - Cambio de contraseña / seguridad
13
14 ## ■ 3. Configuración Académica
15 - Institución (datos, logo, jornadas)
16 - Periodo lectivo / Año escolar
17 - Niveles, cursos y paralelos
18 - Materias / Asignaturas
19
20 ## ■■■ 4. Gestión Docente
21 - Asignación docente-materia-curso
22 - Tutorías (tutor por paralelo)
23
24 ## ■ 5. Gestión Estudiantil
25 - Registro de estudiantes
26 - Asignación a curso/paralelo
27 - Estado del estudiante
28
29 ## ■ 6. Representantes
30 - Registro y vinculación con estudiantes
31 - ■ Mis Representados (vista del representante)
32
33 ## ■ 7. Horarios
34 - Configuración de bloques horarios
35 - Asignación de horario por curso
36 - Visualización de horarios
37
38 ## ■ 8. Asistencia
39 - Registrar asistencia (docente – por clase del día)
40 - Ver asistencias (por curso / por estudiante)
41 - Mi asistencia (vista estudiante)
42 - Calendario de asistencia
43
44 ## ■ 9. Alertas Tempranas ← ■ FALTA CONSTRUIR
45 - Configuración de reglas (X faltas consecutivas / acumuladas)
46 - Alertas generadas y seguimiento
47 - Notificación automática a representantes
48
49 ## ■ 10. Justificaciones
50 - Solicitar justificación (subir documento)
51 - Revisar justificaciones (Inspector/Tutor)
52 - Justificaciones revisadas
53 - Mis justificaciones
54
55 ## ■ 11. Reportes y Estadísticas
56 - Generar reportes (asistencia por curso/estudiante)
57 - Estadísticas visuales
58 - Exportación PDF / Excel
```

59

60 ## ■ 12. Comunicación ← ■ FALTA CONSTRUIR

61 - Mensajería interna

62 - Avisos institucionales

63 - Notificación a representantes

64 - Historial de comunicaciones

65

66 ## ■ 13. Administración del Sistema

67 - RespalDOS (backup / restaurar)

68 - ■ Logs y auditoría del sistema ← ■ FALTA VISTA

69 - ■ Configuración de correo SMTP ← ■ FALTA VISTA

70 - ■ Documentos institucionales ← ■ FALTA

71

72 ■ Lo que FALTA (brechas reales)

73 Del módulo 9 – Sistema de alerta temprana: No existe. No hay controlador, modelo, ni vista para reglas de alerta por fal.

74 Del módulo 12 – Comunicación interna: No existe. No hay mensajería, avisos institucionales ni historial de comunicacione.

75 Del módulo 13 – Incompleto:

76

77 Logs del sistema: Logger.php existe pero sin vista de auditoría

78 Configuración de correo: Mailer.php existe pero sin interfaz de configuración

79 Subida de documentos institucionales: no implementado

80

81 Del módulo 1 – Dashboard: Accesos rápidos y notificaciones visuales en tiempo real son básicos o inexistentes.

8. ArchImp/config/.htaccess

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

9. ArchImp/config/config.php

```
1  <?php
2  // config/config.php
3  // BASE_PATH y BASE_URL vienen de env.php
4  require_once __DIR__ . '/env.php';
5
6  date_default_timezone_set('America/Guayaquil');
7
8  define('EDIT_ATTENDANCE_HOURS', 48);
9
10 // Sesión
11 ini_set('session.gc_maxlifetime', 86400);
12 session_set_cookie_params([
13     'lifetime' => 86400,
14     'path'     => '/',
15     'domain'   => '',
16     'secure'   => (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] !== 'off'),
17     'httponly' => true,
18     'samesite' => 'Lax',
19 ]);
20 session_start();
21
22 // Timeout 30 minutos
23 $inactive_timeout = 1800;
24 if (isset($_SESSION['last_activity']) &&
25     (time() - $_SESSION['last_activity'] > $inactive_timeout)) {
26     session_unset();
27     session_destroy();
28     header('Location: ' . BASE_URL . '/?action=login&timeout=1');
29     exit;
30 }
31 $_SESSION['last_activity'] = time();
32
33 // SMTP
34 define('SMTP_HOST', 'smtp.gmail.com');
35 define('SMTP_PORT', 587);
36 define('SMTP_USER', 'tu-email@gmail.com');
37 define('SMTP_PASS', 'tu-password-app');
38 define('SMTP_FROM', 'noreply@ecuasist.edu.ec');
39 define('SMTP_NAME', 'EcuAsistencia2026');
40
41 require_once BASE_PATH . '/config/database.php';
42 require_once BASE_PATH . '/helpers/Security.php';
```

10. ArchImp/config/database.php

```
1  <?php
2  class Database {
3      private $host = 'localhost';
4      private $db = 'ecuasistencia2026_db';
5      private $user = 'root';
6      private $pass = '';
7      private $charset = 'utf8mb4';
8      private $pdo;
9
10     public function connect() {
11         if ($this->pdo === null) {
12             $dsn = "mysql:host={$this->host};dbname={$this->db};charset={$this->charset}";
13             $options = [
14                 PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
15                 PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
16                 PDO::ATTR_EMULATE_PREPARES => false,
17             ];
18
19             try {
20                 $this->pdo = new PDO($dsn, $this->user, $this->pass, $options);
21             } catch (PDOException $e) {
22                 die('Error de conexión: ' . $e->getMessage());
23             }
24         }
25         return $this->pdo;
26     }
27 }
```

11. ArchImp/config/env.php

```
1  <?php
2  // Detecta automáticamente si estás en local o producción
3  if ($_SERVER['SERVER_NAME'] === 'localhost' || $_SERVER['SERVER_NAME'] === '127.0.0.1') {
4      // LOCAL
5      define('BASE_PATH', __DIR__ . '/../');
6      define('BASE_URL', 'http://localhost/ecuasistencia2026/public');
7  } else {
8      // PRODUCCIÓN
9      define('BASE_PATH', '/home/ecuasys/ecuasistencia');
10     define('BASE_URL', 'https://www.ecuasys.com');
11 }
```

12. ArchImp/controllers/AcademicController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Course.php';
4  require_once BASE_PATH . '/models/Subject.php';
5  require_once BASE_PATH . '/models/SchoolYear.php';
6  require_once BASE_PATH . '/models/Shift.php';
7  require_once BASE_PATH . '/models/User.php';
8
9  class AcademicController {
10     private $courseModel;
11     private $subjectModel;
12     private $schoolYearModel;
13     private $shiftModel;
14     private $userModel;
15     private $db;
16
17     public function __construct() {
18         Security::requireLogin();
19         if (!Security::hasRole('autoridad')) {
20             die('Acceso denegado');
21         }
22
23         $this->db = new Database();
24         $this->courseModel = new Course($this->db);
25         $this->subjectModel = new Subject($this->db);
26         $this->schoolYearModel = new SchoolYear($this->db);
27         $this->shiftModel = new Shift($this->db);
28         $this->userModel = new User($this->db);
29     }
30
31     public function index() {
32         $courses = $this->courseModel->getAll();
33         $subjects = $this->subjectModel->getAll();
34         $schoolYears = $this->schoolYearModel->getAll();
35         $shifts = $this->shiftModel->getAll();
36
37         include BASE_PATH . '/views/academic/index.php';
38     }
39
40     public function createCourse() {
41         if ($_SERVER['REQUEST_METHOD'] !== 'POST') return;
42
43         // Guardar datos del form en sesión para repoblar si hay error
44         $_SESSION['course_form'] = [
45             'education_type' => $_POST['education_type'] ?? '',
46             'grade_level' => $_POST['grade_level'] ?? '',
47             'specialty' => $_POST['specialty'] ?? '',
48             'carrera' => $_POST['carrera'] ?? '',
49             'parallel' => $_POST['parallel'] ?? '',
50             'shift_id' => $_POST['shift_id'] ?? '',
51             'name' => $_POST['name'] ?? '',
52         ];
53
54         $activeYear = $this->schoolYearModel->getActive();
55         if (!$activeYear) {
56             header('Location: ?action=academic&error=no_active_year'); exit;
57         }
58     }
```



```

59     $gradeLevel = html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8');
60     $parallel    = Security::sanitize($_POST['parallel']);
61     $shiftId     = (int)$_POST['shift_id'];
62
63     // Validar curso duplicado
64     $pdo = $this->db->connect();
65     $stmtCheck = $pdo->prepare(
66         "SELECT COUNT(*) as cnt FROM courses
67         WHERE institution_id=:iid AND school_year_id=:syid
68         AND grade_level=:gl AND parallel=:par AND shift_id=:sid"
69     );
70     $stmtCheck->execute([':iid'=>$_SESSION['institution_id'], ':syid'=>$activeYear['id'],
71         ':gl'=>$gradeLevel, ':par'=>$parallel, ':sid'=>$shiftId]);
72     if ($stmtCheck->fetch()['cnt'] > 0) {
73         header('Location: ?action=academic&error=course_duplicate'); exit;
74     }
75
76     $data = [
77         ':institution_id' => $_SESSION['institution_id'],
78         ':school_year_id' => $activeYear['id'],
79         ':name'           => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
80         ':grade_level'    => $gradeLevel,
81         ':parallel'       => $parallel,
82         ':shift_id'       => $shiftId,
83     ];
84     $this->courseModel->create($data);
85     $courseId = (int)$this->db->connect()->lastInsertId();
86
87     // Auto-carga malla curricular
88     $malla = $this->getMallaCurricular();
89     $nuevas = 0;
90     if (isset($malla[$gradeLevel]) && $courseId > 0) {
91         foreach ($malla[$gradeLevel] as $nombre) {
92             $stmtB = $pdo->prepare("SELECT id FROM subjects WHERE institution_id=:iid AND LOWER(TRIM(name))=LOWER.");
93             $stmtB->execute([':iid'=>$_SESSION['institution_id'], ':name'=>$nombre]);
94             $existe = $stmtB->fetch();
95             if ($existe) {
96                 $sid = (int)$existe['id'];
97             } else {
98                 $codigo = strtoupper(substr(preg_replace('/[^a-zA-Z]/', '', iconv('UTF-8', 'ASCII//TRANSLIT', $nombre)), 0, 3));
99                 $this->subjectModel->create([':institution_id'=>$_SESSION['institution_id'], ':name'=>$nombre]);
100                 $sid = (int)$pdo->lastInsertId();
101                 $nuevas++;
102             }
103             $this->linkSubjectToCourse($courseId, $sid);
104         }
105     }
106
107     // Éxito: limpiar datos guardados
108     unset($_SESSION['course_form']);
109     header('Location: ?action=academic&course_success=1&subjects_loaded='.$nuevas); exit;
110 }
111
112 public function createSubject() {
113     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
114         $data = [
115             ':institution_id' => $_SESSION['institution_id'],
116             ':name'           => Security::sanitize($_POST['name']),
117             ':code'           => Security::sanitize($_POST['code'])

```

```

118         ];
119
120         $this->subjectModel->create($data);
121         header('Location: ?action=academic&subject_success=1');
122         exit;
123     }
124 }
125
126 // =====
127 // CRUD CURSOS
128 // =====
129
130 public function editCourse() {
131     $courseId = (int)$_GET['id'];
132     $course = $this->courseModel->findById($courseId);
133
134     if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
135         header('Location: ?action=academic&error=course_not_found');
136         exit;
137     }
138
139     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
140         $shifts = $this->shiftModel->getAll();
141         include BASE_PATH . '/views/academic/course_edit.php';
142         return;
143     }
144
145     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
146         $data = [
147             'id' => $courseId,
148             'name' => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
149             'grade_level' => html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8'),
150             'parallel' => Security::sanitize($_POST['parallel']),
151             'shift_id' => (int)$_POST['shift_id']
152         ];
153
154         if ($this->courseModel->update($data)) {
155             header('Location: ?action=academic&course_updated=1');
156             exit;
157         } else {
158             $errors[] = "Error al actualizar el curso";
159             $shifts = $this->shiftModel->getAll();
160             include BASE_PATH . '/views/academic/course_edit.php';
161         }
162     }
163 }
164
165 public function deleteCourse() {
166     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
167         $courseId = (int)$_POST['course_id'];
168         $course = $this->courseModel->findById($courseId);
169
170         if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
171             header('Location: ?action=academic&error=course_not_found');
172             exit;
173         }
174
175         // Verificar si tiene estudiantes matriculados
176         $students = $this->courseModel->getEnrolledStudents($courseId);

```

[illegible]

[illegible]

[illegible]

[illegible]

```

413         header('Location: ?action=academic&error=subject_not_found');
414         exit;
415     }
416
417     // Verificar si tiene asignaciones docentes
418     $db = new Database();
419     $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM teacher_assignments WHERE subject_id = .");
420     $stmt->execute([':id' => $subjectId]);
421     $result = $stmt->fetch();
422
423     if ($result['count'] > 0) {
424         header('Location: ?action=academic&error=subject_has_assignments');
425         exit;
426     }
427
428     if ($this->subjectModel->delete($subjectId)) {
429         header('Location: ?action=academic&subject_deleted=1');
430         exit;
431     } else {
432         header('Location: ?action=academic&error=delete_failed');
433         exit;
434     }
435 }
436 }
437
438 public function enrollStudents() {
439     $activeYear = $this->schoolYearModel->getActive();
440     $courses = $this->courseModel->getAll();
441     $availableStudents = $this->userModel->getStudentsNotEnrolled($activeYear['id']);
442     $allStudents = $this->userModel->getByRole('estudiante');
443
444     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
445         $courseId = (int)$_POST['course_id'];
446         $studentIds = $_POST['student_ids'] ?? [];
447
448         $enrolled = 0;
449         $errors = 0;
450
451         foreach ($studentIds as $studentId) {
452             if ($this->courseModel->enrollStudent($courseId, (int)$studentId, $activeYear['id'])) {
453                 $enrolled++;
454             } else {
455                 $errors++;
456             }
457         }
458
459         header('Location: ?action=enroll_students&enrolled=' . $enrolled . '&errors=' . $errors);
460         exit;
461     }
462
463     include BASE_PATH . '/views/academic/enroll.php';
464 }
465
466 public function unenrollStudent() {
467     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
468         $studentId = (int)$_POST['student_id'];
469         $activeYear = $this->schoolYearModel->getActive();
470
471         if (!$activeYear) {

```

```

472         header('Location: ?action=enroll_students&error=no_active_year');
473         exit;
474     }
475
476     // Verificar que el estudiante existe y está matriculado
477     $course = $this->userModel->getStudentCourse($studentId, $activeYear['id']);
478
479     if (!$course) {
480         header('Location: ?action=enroll_students&error=not_enrolled');
481         exit;
482     }
483
484     // Retirar estudiante
485     if ($this->courseModel->unenrollStudent($studentId, $activeYear['id'])) {
486         header('Location: ?action=enroll_students&unenrolled=1');
487         exit;
488     } else {
489         header('Location: ?action=enroll_students&error=unenroll_failed');
490         exit;
491     }
492 }
493 }
494
495 public function viewCourseStudents() {
496     $courseId = (int)($_GET['course_id'] ?? 0);
497
498     if (!$courseId) {
499         header('Location: ?action=academic');
500         exit;
501     }
502
503     $course = $this->courseModel->getAll();
504     $course = array_filter($course, fn($c) => $c['id'] == $courseId);
505     $course = reset($course);
506
507     $students = $this->courseModel->getEnrolledStudents($courseId);
508
509     include BASE_PATH . '/views/academic/course_students.php';
510 }
511
512 // =====
513 // CRUD AÑOS LECTIVOS
514 // =====
515
516 public function createSchoolYear() {
517     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
518         include BASE_PATH . '/views/academic/school_year_create.php';
519         return;
520     }
521
522     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
523         $errors = [];
524
525         // Validaciones
526         if (empty($_POST['name'])) {
527             $errors[] = "El nombre es obligatorio";
528         }
529         if (empty($_POST['start_date'])) {
530             $errors[] = "La fecha de inicio es obligatoria";

```



```

531     }
532     if (empty($_POST['end_date'])) {
533         $errors[] = "La fecha de fin es obligatoria";
534     }
535
536     if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {
537         if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
538             $errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
539         }
540
541         // Verificar solapamiento de fechas
542         if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'])) {
543             $errors[] = "Ya existe un año lectivo con fechas que se solapan";
544         }
545     }
546
547     if (!empty($errors)) {
548         include BASE_PATH . '/views/academic/school_year_create.php';
549         return;
550     }
551
552     // Crear año lectivo
553     $data = [
554         'institution_id' => $_SESSION['institution_id'],
555         'name' => Security::sanitize($_POST['name']),
556         'start_date' => $_POST['start_date'],
557         'end_date' => $_POST['end_date'],
558         'is_active' => isset($_POST['is_active']) ? 1 : 0
559     ];
560
561     // Si se marca como activo, desactivar los demás ANTES de insertar
562     if ($data['is_active']) {
563         $this->schoolYearModel->activate(0);
564     }
565
566     // create() ahora devuelve el ID insertado (o 0 si falla)
567     $newId = $this->schoolYearModel->create($data);
568
569     if ($newId > 0) {
570         // Activar el nuevo año lectivo con el ID real
571         if ($data['is_active']) {
572             $this->schoolYearModel->activate($newId);
573         }
574         header('Location: ?action=academic&sy_created=1');
575         exit;
576     } else {
577         $errors[] = "Error al crear el año lectivo";
578         include BASE_PATH . '/views/academic/school_year_create.php';
579     }
580 }
581
582
583 public function editSchoolYear() {
584     $yearId = (int)$_GET['id'];
585     $year = $this->schoolYearModel->findById($yearId);
586
587     if (!$year) {
588         header('Location: ?action=academic&error=year_not_found');
589         exit;

```

```

590     }
591
592     // Verificar institución
593     if ($year['institution_id'] != $_SESSION['institution_id']) {
594         die('Acceso denegado');
595     }
596
597     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
598         include BASE_PATH . '/views/academic/school_year_edit.php';
599         return;
600     }
601
602     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
603         $errors = [];
604
605         // Validaciones
606         if (empty($_POST['name'])) {
607             $errors[] = "El nombre es obligatorio";
608         }
609         if (empty($_POST['start_date'])) {
610             $errors[] = "La fecha de inicio es obligatoria";
611         }
612         if (empty($_POST['end_date'])) {
613             $errors[] = "La fecha de fin es obligatoria";
614         }
615
616         if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {
617             if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
618                 $errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
619             }
620
621             // Verificar solapamiento (excluyendo el actual)
622             if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'], $yearId)) {
623                 $errors[] = "Ya existe otro año lectivo con fechas que se solapan";
624             }
625         }
626
627         if (!empty($errors)) {
628             include BASE_PATH . '/views/academic/school_year_edit.php';
629             return;
630         }
631
632         // Actualizar año lectivo
633         $data = [
634             'id' => $yearId,
635             'name' => Security::sanitize($_POST['name']),
636             'start_date' => $_POST['start_date'],
637             'end_date' => $_POST['end_date']
638         ];
639
640         if ($this->schoolYearModel->update($data)) {
641             header('Location: ?action=academic&sy_updated=1');
642             exit;
643         } else {
644             $errors[] = "Error al actualizar el año lectivo";
645             include BASE_PATH . '/views/academic/school_year_edit.php';
646         }
647     }
648 }

```

```

649
650 public function deleteSchoolYear() {
651     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
652         $yearId = (int)$_POST['year_id'];
653         $year = $this->schoolYearModel->findById($yearId);
654
655         if (!$year) {
656             header('Location: ?action=academic&error=year_not_found');
657             exit;
658         }
659
660         // Verificar institución
661         if ($year['institution_id'] != $_SESSION['institution_id']) {
662             die('Acceso denegado');
663         }
664
665         // No permitir eliminar año activo
666         if ($year['is_active'] == 1) {
667             header('Location: ?action=academic&error=cannot_delete_active');
668             exit;
669         }
670
671         if ($this->schoolYearModel->delete($yearId)) {
672             header('Location: ?action=academic&sy_deleted=1');
673             exit;
674         } else {
675             header('Location: ?action=academic&error=has_courses');
676             exit;
677         }
678     }
679 }
680
681 public function activateSchoolYear() {
682     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
683         $yearId = (int)$_POST['year_id'];
684         $year = $this->schoolYearModel->findById($yearId);
685
686         if (!$year) {
687             header('Location: ?action=academic&error=year_not_found');
688             exit;
689         }
690
691         // Verificar institución
692         if ($year['institution_id'] != $_SESSION['institution_id']) {
693             die('Acceso denegado');
694         }
695
696         if ($this->schoolYearModel->activate($yearId)) {
697             header('Location: ?action=academic&sy_activated=1');
698             exit;
699         } else {
700             header('Location: ?action=academic&error=activate_failed');
701             exit;
702         }
703     }
704 }
705
706 public function deactivateSchoolYear() {
707     if ($_SERVER['REQUEST_METHOD'] === 'POST') {

```

```
708     $yearId = (int)$_POST['year_id'];
709     $year = $this->schoolYearModel->findById($yearId);
710
711     if (!$year) {
712         header('Location: ?action=academic&error=year_not_found');
713         exit;
714     }
715
716     // Verificar institución
717     if ($year['institution_id'] != $_SESSION['institution_id']) {
718         die('Acceso denegado');
719     }
720
721     if ($this->schoolYearModel->deactivate($yearId)) {
722         header('Location: ?action=academic&sy_deactivated=1');
723         exit;
724     } else {
725         header('Location: ?action=academic&error=deactivate_failed');
726         exit;
727     }
728 }
729 }
730 }
```

13. ArchImp/controllers/AssignmentController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/TeacherAssignment.php';
4  require_once BASE_PATH . '/models/Course.php';
5  require_once BASE_PATH . '/models/Subject.php';
6  require_once BASE_PATH . '/models/SchoolYear.php';
7  require_once BASE_PATH . '/models/User.php';
8
9  class AssignmentController {
10     private $assignmentModel;
11     private $courseModel;
12     private $subjectModel;
13     private $schoolYearModel;
14     private $userModel;
15
16     public function __construct() {
17         $db = new Database();
18         $this->assignmentModel = new TeacherAssignment($db);
19         $this->courseModel = new Course($db);
20         $this->subjectModel = new Subject($db);
21         $this->schoolYearModel = new SchoolYear($db);
22         $this->userModel = new User($db);
23     }
24
25     public function index() {
26         Security::requireLogin();
27         if (!Security::hasRole('autoridad')) {
28             die('Acceso denegado');
29         }
30
31         $courses = $this->courseModel->getAll();
32         $subjects = $this->subjectModel->getAll();
33         $teachers = $this->userModel->getByRole('docente');
34         $activeYear = $this->schoolYearModel->getActive();
35         $assignments = $this->assignmentModel->getAll();
36
37         include BASE_PATH . '/views/assignments/index.php';
38     }
39
40     public function assign() {
41         Security::requireLogin();
42         if (!Security::hasRole('autoridad')) {
43             die('Acceso denegado');
44         }
45
46         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
47             $activeYear = $this->schoolYearModel->getActive();
48
49             $data = [
50                 'teacher_id' => (int)$_POST['teacher_id'],
51                 'course_id' => (int)$_POST['course_id'],
52                 'subject_id' => (int)$_POST['subject_id'],
53                 'school_year_id' => $activeYear['id'],
54                 'is_tutor' => 0
55             ];
56
57             $result = $this->assignmentModel->assign($data);
58         }
```

```

59         if ($result['success']) {
60             header('Location: ?action=assignments&success=1');
61         } else {
62             header('Location: ?action=assignments&error=' . urlencode($result['message']));
63         }
64         exit;
65     }
66 }
67
68 public function setTutor() {
69     Security::requireLogin();
70     if (!Security::hasRole('autoridad')) {
71         die('Acceso denegado');
72     }
73
74     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
75         $activeYear = $this->schoolYearModel->getActive();
76         $courseId = (int)$_POST['course_id'];
77         $teacherId = (int)$_POST['teacher_id'];
78
79         $result = $this->assignmentModel->setTutor($courseId, $teacherId, $activeYear['id']);
80
81         if ($result['success']) {
82             header('Location: ?action=tutor_management&tutor_success=1');
83         } else {
84             header('Location: ?action=tutor_management&tutor_error=' . urlencode($result['message']));
85         }
86         exit;
87     }
88 }
89
90 public function removeTutor() {
91     Security::requireLogin();
92     if (!Security::hasRole('autoridad')) {
93         die('Acceso denegado');
94     }
95
96     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
97         $activeYear = $this->schoolYearModel->getActive();
98         $courseId = (int)$_POST['course_id'];
99
100         $db = new Database();
101         $sql = "UPDATE teacher_assignments
102             SET is_tutor = 0
103             WHERE course_id = :course_id
104             AND school_year_id = :school_year_id";
105
106         $stmt = $db->connect()->prepare($sql);
107         $stmt->execute([
108             ':course_id' => $courseId,
109             ':school_year_id' => $activeYear['id']
110         ]);
111
112         header('Location: ?action=tutor_management&tutor_removed=1');
113         exit;
114     }
115 }
116
117 public function remove() {

```

```

118     Security::requireLogin();
119     if (!Security::hasRole('autoridad')) {
120         die('Acceso denegado');
121     }
122
123     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
124         $assignmentId = (int)$_POST['assignment_id'];
125         $result = $this->assignmentModel->remove($assignmentId);
126
127         if ($result['success']) {
128             header('Location: ?action=assignments&removed=1');
129         } else {
130             header('Location: ?action=assignments&error=' . urlencode($result['message']));
131         }
132         exit;
133     }
134 }
135
136 public function viewByCourse() {
137     Security::requireLogin();
138     if (!Security::hasRole('autoridad')) {
139         die('Acceso denegado');
140     }
141
142     $courseId = (int)($_GET['course_id'] ?? 0);
143
144     if (!$courseId) {
145         header('Location: ?action=assignments');
146         exit;
147     }
148
149     $courses = $this->courseModel->getAll();
150     $course = array_filter($courses, fn($c) => $c['id'] == $courseId);
151     $course = reset($course);
152
153     $activeYear = $this->schoolYearModel->getActive();
154     $assignments = $this->assignmentModel->getByCourse($courseId);
155     $tutor = $this->assignmentModel->getTutorByCourse($courseId, $activeYear['id']);
156
157     include BASE_PATH . '/views/assignments/view_course.php';
158 }
159
160 public function getCourseTeachers() {
161     if (!isset($_SESSION['user_id'])) {
162         header('Content-Type: application/json');
163         echo json_encode(['error' => 'No autenticado']);
164         exit;
165     }
166
167     $courseId = (int)($_GET['course_id'] ?? 0);
168
169     if (!$courseId) {
170         header('Content-Type: application/json');
171         echo json_encode([]);
172         exit;
173     }
174
175     // Obtener año lectivo activo
176     $db = new Database();

```

```

177     $sqlYear = "SELECT id FROM school_years WHERE is_active = 1 LIMIT 1";
178     $stmtYear = $db->connect()->query($sqlYear);
179     $activeYear = $stmtYear->fetch();
180
181     if (!$activeYear) {
182         header('Content-Type: application/json');
183         echo json_encode([]);
184         exit;
185     }
186
187     // Solo docentes del curso que:
188     // 1. NO son tutores de otro curso
189     // 2. NO son tutores del curso actual
190     $sql = "SELECT DISTINCT ta.teacher_id,
191             CONCAT(u.last_name, ' ', u.first_name) as teacher_name
192             FROM teacher_assignments ta
193             INNER JOIN users u ON ta.teacher_id = u.id
194             WHERE ta.course_id = :course_id
195             AND ta.teacher_id NOT IN (
196                 SELECT teacher_id
197                 FROM teacher_assignments
198                 WHERE is_tutor = 1
199                 AND school_year_id = :school_year_id
200             )
201             ORDER BY u.last_name, u.first_name";
202
203     $stmt = $db->connect()->prepare($sql);
204     $stmt->execute([
205         ':course_id' => $courseId,
206         ':school_year_id' => $activeYear['id']
207     ]);
208     $teachers = $stmt->fetchAll(PDO::FETCH_ASSOC);
209
210     header('Content-Type: application/json');
211     echo json_encode($teachers);
212     exit;
213 }
214
215 public function tutorManagement() {
216     $courses = $this->courseModel->getAll();
217     $assignments = $this->assignmentModel->getAll();
218
219     include BASE_PATH . '/views/assignments/tutor.php';
220 }
221
222 public function checkCourseTutor() {
223     if (!isset($_SESSION['user_id'])) {
224         header('Content-Type: application/json');
225         echo json_encode(null);
226         exit;
227     }
228
229     $courseId = (int)($_GET['course_id'] ?? 0);
230
231     if (!$courseId) {
232         header('Content-Type: application/json');
233         echo json_encode(null);
234         exit;
235     }

```



```

236
237     $db = new Database();
238     require_once BASE_PATH . '/models/SchoolYear.php';
239     $yearModel = new SchoolYear($db);
240     $activeYear = $yearModel->getActive();
241
242     $sql = "SELECT CONCAT(u.last_name, ' ', u.first_name) as tutor_name
243           FROM teacher_assignments ta
244           INNER JOIN users u ON ta.teacher_id = u.id
245           WHERE ta.course_id = :course_id
246           AND ta.school_year_id = :school_year_id
247           AND ta.is_tutor = 1
248           LIMIT 1";
249
250     $stmt = $db->connect()->prepare($sql);
251     $stmt->execute([
252         ':course_id' => $courseId,
253         ':school_year_id' => $activeYear['id']
254     ]);
255
256     $tutor = $stmt->fetch(PDO::FETCH_ASSOC);
257
258     header('Content-Type: application/json');
259     echo json_encode($tutor);
260     exit;
261 }
262 }

```

14. ArchImp/controllers/AttendanceController.php

```
1  <?php
2  // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3  // controllers/AttendanceController.php
4
5  require_once BASE_PATH . '/config/config.php';
6  require_once BASE_PATH . '/models/Attendance.php';
7  require_once BASE_PATH . '/models/Course.php';
8  require_once BASE_PATH . '/models/Subject.php';
9  require_once BASE_PATH . '/models/SchoolYear.php';
10 require_once BASE_PATH . '/models/Shift.php';
11 require_once BASE_PATH . '/models/Notification.php';
12
13 class AttendanceController {
14     private $attendanceModel;
15     private $courseModel;
16     private $subjectModel;
17     private $schoolYearModel;
18     private $shiftModel;
19     private $notificationModel;
20
21     public function __construct() {
22         Security::requireLogin();
23         $db = new Database();
24
25         $this->attendanceModel = new Attendance($db);
26         $this->courseModel = new Course($db);
27         $this->subjectModel = new Subject($db);
28         $this->schoolYearModel = new SchoolYear($db);
29         $this->shiftModel = new Shift($db);
30         $this->notificationModel = new Notification($db);
31     }
32
33     private function isWithin48BusinessHours($date) {
34         $targetDate = new DateTime($date);
35         $today = new DateTime();
36         $today->setTime(0, 0, 0);
37
38         if ($targetDate->format('Y-m-d') === $today->format('Y-m-d')) {
39             return true;
40         }
41         if ($targetDate > $today) {
42             return false;
43         }
44
45         $businessHours = 0;
46         $current = clone $targetDate;
47
48         while ($current < $today) {
49             $current->modify('+1 day');
50             $dayOfWeek = (int)$current->format('N');
51             if ($dayOfWeek < 6) {
52                 $businessHours += 24;
53             }
54         }
55
56         return $businessHours <= EDIT_ATTENDANCE_HOURS;
57     }
58 }
```

```

59     private function getMaxEditDate() {
60         $today = new DateTime();
61         $businessHoursNeeded = EDIT_ATTENDANCE_HOURS;
62         $current = clone $today;
63
64         while ($businessHoursNeeded > 0) {
65             $current->modify('+1 day');
66             $dayOfWeek = (int)$current->format('N');
67             if ($dayOfWeek < 6) {
68                 $businessHoursNeeded -= 24;
69             }
70         }
71
72         return $current->format('Y-m-d');
73     }
74
75     public function register() {
76         if (!Security::hasRole(['docente', 'autoridad'])) {
77             die('Acceso denegado');
78         }
79
80         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
81             $activeYear = $this->schoolYearModel->getActive();
82             $date = $_POST['date'];
83             $today = date('Y-m-d');
84
85             if (strtotime($date) > strtotime($today)) {
86                 header('Location: ?action=attendance_register&error=future');
87                 exit;
88             }
89
90             if (!$this->isWithin48BusinessHours($date)) {
91                 header('Location: ?action=attendance_register&error=toolate');
92                 exit;
93             }
94
95             $scheduleId = (int)$_POST['schedule_id'];
96
97             // Obtener datos del horario incluyendo nombre de asignatura
98             $db = new Database();
99             $pdo = $db->connect();
100             $sql = "SELECT cs.*, s.name as subject_name
101                 FROM class_schedule cs
102                 INNER JOIN subjects s ON cs.subject_id = s.id
103                 WHERE cs.id = :id";
104             $stmt = $pdo->prepare($sql);
105             $stmt->execute([':id' => $scheduleId]);
106             $scheduleData = $stmt->fetch();
107
108             $subjectName = $scheduleData['subject_name'] ?? 'clase';
109             $dateFormatted = date('d/m/Y', strtotime($date));
110
111             foreach ($_POST as $key => $value) {
112                 if (strpos($key, 'status_') === 0) {
113                     $studentId = (int)str_replace('status_', '', $key);
114                     $status = Security::sanitize($value);
115                     $observation = Security::sanitize($_POST['obs_' . $studentId] ?? '');
116
117                     $data = [

```

```

118         ':student_id'      => $studentId,
119         ':course_id'      => $scheduleData['course_id'],
120         ':subject_id'     => $scheduleData['subject_id'],
121         ':teacher_id'     => $_SESSION['user_id'],
122         ':school_year_id' => $activeYear['id'],
123         ':shift_id'       => 1,
124         ':date'           => $date,
125         ':hour_period'    => $scheduleData['period_number'] . 'ra hora',
126         ':status'         => $status,
127         ':observation'    => $observation
128     ];
129
130     $this->attendanceModel->create($data);
131
132     // Notificar al estudiante si fue marcado ausente
133     // El enlace lleva directo a su asistencia para que pueda justificar
134     if ($status === 'ausente') {
135         $this->notificationModel->create(
136             $studentId,
137             '■ Ausencia registrada',
138             "Se registró una ausencia el {$dateFormatted} en {$subjectName}. Puedes justificarla.",
139             'ausente',
140             '?action=my_attendance'
141         );
142     }
143
144     // Notificar al estudiante si fue marcado con tardanza
145     if ($status === 'tardanza') {
146         $this->notificationModel->create(
147             $studentId,
148             '■ Tardanza registrada',
149             "Se registró una tardanza el {$dateFormatted} en {$subjectName}.",
150             'warning',
151             '?action=my_attendance'
152         );
153     }
154 }
155 }
156
157 header('Location: ?action=attendance_register&success=1');
158 exit;
159 }
160
161 // Obtener clases del docente para hoy
162 $activeYear = $this->schoolYearModel->getActive();
163
164 $db = new Database();
165 require_once BASE_PATH . '/models/ClassSchedule.php';
166 $scheduleModel = new ClassSchedule($db);
167
168 $todayClasses = $scheduleModel->getTeacherScheduleToday($_SESSION['user_id'], $activeYear['id']);
169 $minDate      = $this->calculateMinDate();
170 $maxEditDate  = $this->getMaxEditDate();
171
172 include BASE_PATH . '/views/attendance/register.php';
173 }
174
175 private function getTeacherCourses($teacherId) {
176     $sql = "SELECT DISTINCT c.*, sh.name as shift_name

```

```

177         FROM courses c
178         INNER JOIN teacher_assignments ta ON c.id = ta.course_id
179         INNER JOIN shifts sh ON c.shift_id = sh.id
180         WHERE ta.teacher_id = :teacher_id
181         ORDER BY c.name";
182
183     $db = new Database();
184     $stmt = $db->connect()->prepare($sql);
185     $stmt->execute([':teacher_id' => $teacherId]);
186     return $stmt->fetchAll();
187 }
188
189 private function calculateMinDate() {
190     $date = new DateTime();
191     $hoursBack = 0;
192
193     while ($hoursBack < 48) {
194         $date->modify('-1 day');
195         $dayOfWeek = $date->format('N');
196         if ($dayOfWeek >= 1 && $dayOfWeek <= 5) {
197             $hoursBack += 24;
198         }
199     }
200
201     return $date->format('Y-m-d');
202 }
203
204 public function getStudents() {
205     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
206         $courseId = (int)$_POST['course_id'];
207         $students = $this->courseModel->getStudents($courseId);
208         header('Content-Type: application/json');
209         echo json_encode($students);
210         exit;
211     }
212 }
213
214 public function view() {
215     if (!Security::hasRole(['docente', 'inspector', 'autoridad'])) {
216         die('Acceso denegado');
217     }
218
219     $courses = $this->courseModel->getAll();
220     $attendances = [];
221
222     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
223         $courseId = (int)$_POST['course_id'];
224         $date = $_POST['date'];
225         $attendances = $this->attendanceModel->getByCourse($courseId, $date);
226     }
227
228     include BASE_PATH . '/views/attendance/view.php';
229 }
230
231 public function myAttendance() {
232     if (!Security::hasRole('estudiante')) {
233         die('Acceso denegado');
234     }
235 }

```

```

236     $studentId    = $_SESSION['user_id'];
237     $attendances = $this->attendanceModel->getByStudent($studentId);
238
239     include BASE_PATH . '/views/attendance/my_attendance.php';
240 }
241
242 public function calendar() {
243     if (!Security::hasRole(['docente', 'autoridad', 'inspector'])) {
244         die('Acceso denegado');
245     }
246
247     $courses = $this->courseModel->getAll();
248
249     if (isset($_GET['course_id'])) {
250         $courseId = (int)$_GET['course_id'];
251         $month     = $_GET['month'] ?? date('Y-m');
252
253         list($year, $monthNum) = explode('-', $month);
254
255         $monthNames = [
256             '01' => 'Enero',    '02' => 'Febrero',    '03' => 'Marzo',
257             '04' => 'Abril',    '05' => 'Mayo',      '06' => 'Junio',
258             '07' => 'Julio',    '08' => 'Agosto',   '09' => 'Septiembre',
259             '10' => 'Octubre',  '11' => 'Noviembre', '12' => 'Diciembre'
260         ];
261
262         $monthName = $monthNames[$monthNum];
263         $prevMonth = date('Y-m', strtotime($month . '-01 -1 month'));
264         $nextMonth = date('Y-m', strtotime($month . '-01 +1 month'));
265         $firstDay  = date('w', strtotime($year . '-' . $monthNum . '-01'));
266         $daysInMonth = date('t', strtotime($year . '-' . $monthNum . '-01'));
267
268         $calendarDays = [];
269
270         for ($i = 0; $i < $firstDay; $i++) {
271             $calendarDays[] = ['day' => '', 'classes' => ''];
272         }
273
274         for ($day = 1; $day <= $daysInMonth; $day++) {
275             $date = $year . '-' . $monthNum . '-' . str_pad($day, 2, '0', STR_PAD_LEFT);
276             $dayOfWeek = date('w', strtotime($date));
277             $classes = [];
278
279             if ($date == date('Y-m-d')) $classes[] = 'today';
280             if ($dayOfWeek == 0 || $dayOfWeek == 6) $classes[] = 'weekend';
281
282             $attendance = $this->attendanceModel->getDayStats($courseId, $date);
283
284             if ($attendance && $attendance['total'] > 0) {
285                 $classes[] = 'has-attendance';
286             }
287
288             $calendarDays[] = [
289                 'day' => $day,
290                 'classes' => implode(' ', $classes),
291                 'attendance' => $attendance
292             ];
293         }
294     }

```

```

295         include BASE_PATH . '/views/attendance/calendar.php';
296     } else {
297         include BASE_PATH . '/views/attendance/calendar.php';
298     }
299 }
300
301 public function getCourseSubjects() {
302     if (!isset($_SESSION['user_id'])) {
303         header('Content-Type: application/json');
304         echo json_encode(['error' => 'No autenticado']);
305         exit;
306     }
307
308     $courseId = (int)($_GET['course_id'] ?? 0);
309
310     if (!$courseId) {
311         header('Content-Type: application/json');
312         echo json_encode([]);
313         exit;
314     }
315
316     $sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name
317           FROM teacher_assignments ta
318           INNER JOIN subjects s ON ta.subject_id = s.id
319           WHERE ta.course_id = :course_id
320           ORDER BY s.name";
321
322     $db = new Database();
323     $stmt = $db->connect()->prepare($sql);
324     $stmt->execute([':course_id' => $courseId]);
325     $subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);
326
327     header('Content-Type: application/json');
328     echo json_encode($subjects);
329     exit;
330 }
331
332 public function getTeacherCourseSubjects() {
333     if (!isset($_SESSION['user_id'])) {
334         header('Content-Type: application/json');
335         echo json_encode(['error' => 'No autenticado']);
336         exit;
337     }
338
339     $courseId = (int)($_GET['course_id'] ?? 0);
340
341     if (!$courseId) {
342         header('Content-Type: application/json');
343         echo json_encode([]);
344         exit;
345     }
346
347     if (Security::hasRole('autoridad')) {
348         $sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name
349               FROM teacher_assignments ta
350               INNER JOIN subjects s ON ta.subject_id = s.id
351               WHERE ta.course_id = :course_id
352               ORDER BY s.name";
353         $params = [':course_id' => $courseId];

```

```

354     } else {
355         $sql      = "SELECT DISTINCT ta.subject_id, s.name as subject_name
356                     FROM teacher_assignments ta
357                     INNER JOIN subjects s ON ta.subject_id = s.id
358                     WHERE ta.course_id = :course_id
359                     AND ta.teacher_id = :teacher_id
360                     ORDER BY s.name";
361         $params = [':course_id' => $courseId, ':teacher_id' => $_SESSION['user_id']];
362     }
363
364     $db      = new Database();
365     $stmt    = $db->connect()->prepare($sql);
366     $stmt->execute($params);
367     $subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);
368
369     header('Content-Type: application/json');
370     echo json_encode($subjects);
371     exit;
372 }
373
374 public function getScheduleInfo() {
375     if (!isset($_SESSION['user_id'])) {
376         header('Content-Type: application/json');
377         echo json_encode(['error' => 'No autenticado']);
378         exit;
379     }
380
381     $scheduleId = (int)($_GET['schedule_id'] ?? 0);
382
383     if (!$scheduleId) {
384         header('Content-Type: application/json');
385         echo json_encode([]);
386         exit;
387     }
388
389     $sql = "SELECT * FROM class_schedule WHERE id = :id";
390     $db  = new Database();
391     $stmt = $db->connect()->prepare($sql);
392     $stmt->execute([':id' => $scheduleId]);
393     $schedule = $stmt->fetch(PDO::FETCH_ASSOC);
394
395     header('Content-Type: application/json');
396     echo json_encode($schedule);
397     exit;
398 }
399
400 public function getExistingAttendance() {
401     header('Content-Type: application/json');
402
403     if (!isset($_SESSION['user_id'])) {
404         echo json_encode([]);
405         exit;
406     }
407
408     $scheduleId = (int)($_GET['schedule_id'] ?? 0);
409     $date       = $_GET['date'] ?? date('Y-m-d');
410
411     if (!$scheduleId) {
412         echo json_encode([]);

```



```

413         exit;
414     }
415
416     $db = new Database();
417     $pdo = $db->connect();
418     $sql = "SELECT * FROM class_schedule WHERE id = :id";
419     $stmt = $pdo->prepare($sql);
420     $stmt->execute([':id' => $scheduleId]);
421     $schedule = $stmt->fetch();
422
423     if (!$schedule) {
424         echo json_encode([]);
425         exit;
426     }
427
428     $sql2 = "SELECT student_id, status, observation
429             FROM attendances
430             WHERE course_id = :course_id
431             AND subject_id = :subject_id
432             AND date = :date
433             AND hour_period = :hour_period";
434     $stmt2 = $pdo->prepare($sql2);
435     $stmt2->execute([
436         ':course_id' => $schedule['course_id'],
437         ':subject_id' => $schedule['subject_id'],
438         ':date' => $date,
439         ':hour_period' => $schedule['period_number'] . 'ra hora'
440     ]);
441
442     echo json_encode($stmt2->fetchAll(PDO::FETCH_ASSOC));
443     exit;
444 }
445 }

```

15. ArchImp/controllers/AuthController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/User.php';
4  require_once BASE_PATH . '/helpers/Mailer.php';
5
6  class AuthController {
7      private $userModel;
8
9      public function __construct() {
10         $db = new Database();
11         $this->userModel = new User($db);
12     }
13
14     public function register() {
15         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
16             if (!Security::validateToken($_POST['csrf_token'] ?? '')) {
17                 die('Token inválido');
18             }
19
20             $data = [
21                 'institution_id' => 1,
22                 'username' => Security::sanitize($_POST['username']),
23                 'email' => Security::sanitize($_POST['email']),
24                 'password' => $_POST['password'],
25                 'first_name' => Security::sanitize($_POST['first_name']),
26                 'last_name' => Security::sanitize($_POST['last_name']),
27                 'dni' => Security::sanitize($_POST['dni'] ?? ''),
28                 'phone' => Security::sanitize($_POST['phone'] ?? '')
29             ];
30
31             if ($this->userModel->create($data)) {
32                 header('Location: ' . BASE_URL . '/public/index.php?action=login&registered=1');
33                 exit;
34             }
35         }
36
37         include BASE_PATH . '/views/auth/register.php';
38     }
39
40     public function login() {
41         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
42             if (!Security::validateToken($_POST['csrf_token'] ?? '')) {
43                 die('Token inválido');
44             }
45
46             $emailOrUsername = Security::sanitize($_POST['email']);
47             $password = $_POST['password'];
48
49             // Buscar por email o username
50             $user = $this->userModel->findByEmailOrUsername($emailOrUsername);
51
52             if ($user && Security::verifyPassword($password, $user['password'])) {
53                 $_SESSION['user_id'] = $user['id'];
54                 $_SESSION['username'] = $user['username'];
55                 $_SESSION['first_name'] = $user['first_name'];
56                 $_SESSION['last_name'] = $user['last_name'];
57                 $_SESSION['institution_id'] = $user['institution_id'];
58                 $_SESSION['roles'] = $this->userModel->getUserRoles($user['id']);
```

```

59         $_SESSION['is_superadmin'] = !empty($user['is_superadmin']);
60
61         header('Location: ' . BASE_URL . '/public/index.php?action=dashboard');
62         exit;
63     } else {
64         $error = 'Credenciales incorrectas';
65     }
66 }
67
68 include BASE_PATH . '/views/auth/login.php';
69 }
70
71 public function logout() {
72     session_destroy();
73     header('Location: ' . BASE_URL . '/public/index.php?action=login');
74     exit;
75 }
76
77 public function forgotPassword() {
78     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
79         $email = Security::sanitize($_POST['email']);
80         $user = $this->userModel->findByEmail($email);
81
82         if ($user) {
83             $token = bin2hex(random_bytes(32));
84             $this->userModel->setResetToken($email, $token);
85
86             $resetLink = BASE_URL . "/public/index.php?action=reset&token={$token}";
87             $body = "Haga clic aquí para restablecer su contraseña: <a href='{$resetLink}'>{$resetLink}</a>";
88
89             Mailer::send($email, 'Restablecer contraseña', $body);
90         }
91
92         $message = 'Si el correo existe, recibirá un enlace de recuperación';
93     }
94
95     include BASE_PATH . '/views/auth/forgot.php';
96 }
97
98 public function resetPassword() {
99     $token = $_GET['token'] ?? '';
100
101     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
102         $user = $this->userModel->validateResetToken($token);
103
104         if ($user && $_POST['password'] === $_POST['password_confirm']) {
105             $this->userModel->resetPassword($user['id'], $_POST['password']);
106             header('Location: ' . BASE_URL . '/public/index.php?action=login&reset=1');
107             exit;
108         }
109     }
110
111     include BASE_PATH . '/views/auth/reset.php';
112 }
113 }

```

16. ArchImp/controllers/BackupController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/helpers/Backup.php';
4
5  class BackupController {
6      private $backup;
7
8      public function __construct() {
9          Security::requireLogin();
10         if (!Security::hasRole('autoridad')) {
11             die('Acceso denegado');
12         }
13
14         $this->backup = new Backup();
15     }
16
17     public function index() {
18         $backups = $this->backup->getBackups();
19         include BASE_PATH . '/views/backup/index.php';
20     }
21
22     public function create() {
23         $filename = $this->backup->createBackup();
24
25         if ($filename) {
26             header('Location: ?action=backups&success=1&file=' . $filename);
27         } else {
28             header('Location: ?action=backups&error=1');
29         }
30         exit;
31     }
32
33     public function download() {
34         $filename = $_GET['file'] ?? '';
35         $this->backup->downloadBackup($filename);
36     }
37
38     public function cleanup() {
39         $deleted = $this->backup->deleteOldBackups(30);
40         header('Location: ?action=backups&cleanup=' . $deleted);
41         exit;
42     }
43
44     public function delete() {
45         $filename = $_GET['file'] ?? '';
46
47         if ($this->backup->deleteBackup($filename)) {
48             header('Location: ?action=backups&deleted=1');
49         } else {
50             header('Location: ?action=backups&error_delete=1');
51         }
52         exit;
53     }
54 }
```

17. ArchImp/controllers/DashboardController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Attendance.php';
4  require_once BASE_PATH . '/models/User.php';
5  require_once BASE_PATH . '/models/Course.php';
6
7  class DashboardController {
8      private $attendanceModel;
9      private $userModel;
10     private $courseModel;
11
12     public function __construct() {
13         Security::requireLogin();
14
15         $db = new Database();
16         $this->attendanceModel = new Attendance($db);
17         $this->userModel = new User($db);
18         $this->courseModel = new Course($db);
19     }
20
21     public function index() {
22         $stats = $this->getStats();
23         include BASE_PATH . '/views/dashboard/index.php';
24     }
25
26     private function getStats() {
27         $stats = [];
28
29         if (Security::hasRole('autoridad')) {
30             $stats['total_students'] = $this->getTotalByRole('estudiante');
31             $stats['total_teachers'] = $this->getTotalByRole('docente');
32             $stats['total_courses'] = $this->getTotalCourses();
33             $stats['today_attendance'] = $this->getTodayAttendanceStats();
34         }
35
36         if (Security::hasRole('docente')) {
37             $stats['my_courses'] = $this->getTeacherCourses($_SESSION['user_id']);
38             $stats['my_students'] = $this->getTeacherStudentsCount($_SESSION['user_id']);
39             $stats['tutor_course'] = $this->getTutorCourse($_SESSION['user_id']);
40         }
41
42         if (Security::hasRole('estudiante')) {
43             $stats['my_attendance'] = $this->getStudentAttendanceStats($_SESSION['user_id']);
44         }
45
46         if (Security::hasRole('representante')) {
47             $stats['my_children'] = $this->getRepresentativeChildrenCount($_SESSION['user_id']);
48         }
49
50         return $stats;
51     }
52
53     private function getTotalByRole($role) {
54         $sql = "SELECT COUNT(DISTINCT u.id) as total
55             FROM users u
56             INNER JOIN user_roles ur ON u.id = ur.user_id
57             INNER JOIN roles r ON ur.role_id = r.id
58             WHERE r.name = :role AND u.institution_id = :institution_id";
```

```

59
60     $db = new Database();
61     $stmt = $db->connect()->prepare($sql);
62     $stmt->execute([
63         ':role' => $role,
64         ':institution_id' => $_SESSION['institution_id']
65     ]);
66
67     return $stmt->fetch()['total'];
68 }
69
70 private function getTotalCourses() {
71     $sql = "SELECT COUNT(*) as total FROM courses WHERE institution_id = :institution_id";
72
73     $db = new Database();
74     $stmt = $db->connect()->prepare($sql);
75     $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
76
77     return $stmt->fetch()['total'];
78 }
79
80 private function getTodayAttendanceStats() {
81     $sql = "SELECT
82         COUNT(*) as total,
83         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
84         SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente
85     FROM attendances
86     WHERE date = CURDATE()";
87
88     $db = new Database();
89     $stmt = $db->connect()->query($sql);
90     return $stmt->fetch();
91 }
92
93 private function getTeacherCourses($teacherId) {
94     $sql = "SELECT COUNT(DISTINCT course_id) as total
95     FROM teacher_assignments
96     WHERE teacher_id = :teacher_id";
97
98     $db = new Database();
99     $stmt = $db->connect()->prepare($sql);
100     $stmt->execute([':teacher_id' => $teacherId]);
101
102     return $stmt->fetch()['total'];
103 }
104
105 private function getTeacherStudentsCount($teacherId) {
106     $sql = "SELECT COUNT(DISTINCT cs.student_id) as total
107     FROM teacher_assignments ta
108     INNER JOIN course_students cs ON ta.course_id = cs.course_id
109     WHERE ta.teacher_id = :teacher_id";
110
111     $db = new Database();
112     $stmt = $db->connect()->prepare($sql);
113     $stmt->execute([':teacher_id' => $teacherId]);
114
115     return $stmt->fetch()['total'];
116 }
117

```

```

118     private function getStudentAttendanceStats($studentId) {
119         $sql = "SELECT
120             COUNT(*) as total,
121             SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
122             SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
123             SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza
124             FROM attendances
125             WHERE student_id = :student_id
126             AND date >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)";
127
128         $db = new Database();
129         $stmt = $db->connect()->prepare($sql);
130         $stmt->execute([':student_id' => $studentId]);
131
132         return $stmt->fetch();
133     }
134
135     private function getRepresentativeChildrenCount($representativeId) {
136         $sql = "SELECT COUNT(*) as total
137             FROM representatives
138             WHERE representative_id = :representative_id";
139
140         $db = new Database();
141         $stmt = $db->connect()->prepare($sql);
142         $stmt->execute([':representative_id' => $representativeId]);
143
144         return $stmt->fetch()['total'];
145     }
146
147     private function getTutorCourse($teacherId) {
148         $sql = "SELECT c.name as course_name
149             FROM teacher_assignments ta
150             INNER JOIN courses c ON ta.course_id = c.id
151             INNER JOIN school_years sy ON ta.school_year_id = sy.id
152             WHERE ta.teacher_id = :teacher_id
153             AND ta.is_tutor = 1
154             AND sy.is_active = 1
155             LIMIT 1";
156
157         $db = new Database();
158         $stmt = $db->connect()->prepare($sql);
159         $stmt->execute([':teacher_id' => $teacherId]);
160         $row = $stmt->fetch();
161         return $row ? $row['course_name'] : null;
162     }
163 }

```

18. ArchImp/controllers/ExportController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/vendor/autoload.php';
4  require_once BASE_PATH . '/models/Course.php';
5
6  use PhpOffice\PhpSpreadsheet\Spreadsheet;
7  use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
8  use PhpOffice\PhpSpreadsheet\Style\Fill;
9  use PhpOffice\PhpSpreadsheet\Style\Alignment;
10
11 class ExportController {
12     private $courseModel;
13
14     public function __construct() {
15         Security::requireLogin();
16         if (!Security::hasRole(['autoridad', 'docente'])) {
17             die('Acceso denegado');
18         }
19
20         $db = new Database();
21         $this->courseModel = new Course($db);
22     }
23
24     public function exportStudentList() {
25         $courseId = (int)$_GET['course_id'];
26         $students = $this->courseModel->getEnrolledStudents($courseId);
27
28         $courses = $this->courseModel->getAll();
29         $course = array_filter($courses, fn($c) => $c['id'] == $courseId);
30         $course = reset($course);
31
32         $spreadsheet = new Spreadsheet();
33         $sheet = $spreadsheet->getActiveSheet();
34
35         $sheet->setTitle('Nómina');
36
37         // Encabezado
38         $sheet->mergeCells('A1:E1');
39         $sheet->setCellValue('A1', 'NÓMINA DE ESTUDIANTES');
40         $sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
41         $sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);
42
43         $sheet->setCellValue('A3', 'Curso:');
44         $sheet->setCellValue('B3', $course['name'] . ' - ' . $course['shift_name']);
45         $sheet->getStyle('A3')->getFont()->setBold(true);
46
47         $sheet->setCellValue('A4', 'Total:');
48         $sheet->setCellValue('B4', count($students) . ' estudiantes');
49         $sheet->getStyle('A4')->getFont()->setBold(true);
50
51         // Cabecera tabla
52         $headers = ['#', 'Apellidos', 'Nombres', 'Cédula', 'Fecha Matrícula'];
53         $col = 'A';
54         foreach ($headers as $header) {
55             $sheet->setCellValue($col . '6', $header);
56             $sheet->getStyle($col . '6')->getFont()->setBold(true);
57             $sheet->getStyle($col . '6')->getFill()->setFillType(Fill::FILL_SOLID)
58         }
```



```

59         -&gt;getStartColor()-&gt;setRGB('007bff');
60         $sheet-&gt;getStyle($col . '6')-&gt;getFont()-&gt;getColor()-&gt;setRGB('FFFFFF');
61         $col++;
62     }
63
64     // Datos
65     $row = 7;
66     $num = 1;
67     foreach ($students as $student) {
68         $sheet-&gt;setCellValue('A' . $row, $num++);
69         $sheet-&gt;setCellValue('B' . $row, $student['last_name']);
70         $sheet-&gt;setCellValue('C' . $row, $student['first_name']);
71         $sheet-&gt;setCellValue('D' . $row, $student['dni'] ?? '-');
72         $sheet-&gt;setCellValue('E' . $row, date('d/m/Y', strtotime($student['enrollment_date'])));
73         $row++;
74     }
75
76     // Ajustar anchos
77     $sheet-&gt;getColumnDimension('A')-&gt;setWidth(5);
78     $sheet-&gt;getColumnDimension('B')-&gt;setWidth(20);
79     $sheet-&gt;getColumnDimension('C')-&gt;setWidth(20);
80     $sheet-&gt;getColumnDimension('D')-&gt;setWidth(15);
81     $sheet-&gt;getColumnDimension('E')-&gt;setWidth(15);
82
83     $writer = new Xlsx($spreadsheet);
84
85     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
86     header('Content-Disposition: attachment;filename="nomina_' . $course['name'] . '_' . date('YmdHis') . '.xlsx"');
87     header('Cache-Control: max-age=0');
88
89     $writer-&gt;save('php://output');
90     exit;
91 }
92 }

```

19. ArchImp/controllers/InstitutionController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Institution.php';
4  require_once BASE_PATH . '/models/InstitutionShift.php';
5  require_once BASE_PATH . '/models/Shift.php';
6
7  class InstitutionController {
8      private $institutionModel;
9      private $institutionShiftModel;
10     private $shiftModel;
11
12     public function __construct() {
13         Security::requireLogin();
14         if (!Security::hasRole('autoridad')) {
15             die('Acceso denegado');
16         }
17
18         $db = new Database();
19         $this->institutionModel = new Institution($db);
20         $this->institutionShiftModel = new InstitutionShift($db);
21         $this->shiftModel = new Shift($db);
22     }
23
24     public function index() {
25         $institution = $this->institutionModel->getById(1);
26         $allShifts = $this->shiftModel->getAll();
27         $assignedShiftIds = $this->institutionShiftModel->getInstitutionShiftIds(1);
28
29         include BASE_PATH . '/views/institution/index.php';
30     }
31
32     public function update() {
33         if ($_SERVER['REQUEST_METHOD'] !== 'POST') return;
34
35         $current = $this->institutionModel->getById(1);
36         // Mantener logo actual - triple respaldo: BD → hidden → null
37         $logoPath = !empty($current['logo_path'])
38             ? $current['logo_path']
39             : (!empty($_POST['current_logo_path']) ? $_POST['current_logo_path'] : null);
40
41         // Solo procesar si realmente viene un archivo
42         $hasFile = isset($_FILES['logo'])
43             && is_array($_FILES['logo'])
44             && $_FILES['logo']['error'] === UPLOAD_ERR_OK
45             && $_FILES['logo']['size'] > 0;
46
47         if ($hasFile) {
48             $uploadDir = BASE_PATH . '/uploads/institution/';
49
50             // Crear carpeta si no existe (permisos 0775 para hosting compartido)
51             if (!is_dir($uploadDir)) {
52                 mkdir($uploadDir, 0775, true);
53             }
54
55             $ext = strtolower(pathinfo($_FILES['logo']['name'], PATHINFO_EXTENSION));
56             $allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];
57             $maxSize = 2 * 1024 * 1024;
58
```

```

59         if (in_array($ext, $allowed) && $FILES['logo']['size'] <= $maxSize) {
60             $filename = 'logo_' . $_SESSION['institution_id'] . '_' . time() . '.' . $ext;
61             $uploadPath = $uploadDir . $filename;
62
63             if (move_uploaded_file($FILES['logo']['tmp_name'], $uploadPath)) {
64                 // Eliminar logo anterior solo si existe físicamente
65                 if (!empty($current['logo_path'])) {
66                     $oldFile = BASE_PATH . '/' . ltrim($current['logo_path'], '/');
67                     if (file_exists($oldFile)) {
68                         @unlink($oldFile);
69                     }
70                 }
71                 $logoPath = 'uploads/institution/' . $filename;
72             }
73             // Si move_uploaded_file falla: $logoPath mantiene el valor anterior
74         }
75     }
76
77     $data = [
78         'name'          => Security::sanitize($_POST['name']),
79         'address'       => Security::sanitize($_POST['address']),
80         'province'     => Security::sanitize($_POST['province']),
81         'city'         => Security::sanitize($_POST['city']),
82         'phone'        => Security::sanitize($_POST['phone']),
83         'email'        => Security::sanitize($_POST['email']),
84         'director_name'=> Security::sanitize($_POST['director_name']),
85         'amie_code'    => Security::sanitize($_POST['amie_code']),
86         'website'      => Security::sanitize($_POST['website']),
87         'logo_path'    => $logoPath,
88     ];
89
90     $this->institutionModel->update(1, $data);
91     header('Location: ?action=institution&success=1');
92     exit;
93 }
94
95 // Toggle: si está asignada la quita, si no la asigna - responde JSON para AJAX
96 public function toggleShift() {
97     header('Content-Type: application/json');
98
99     if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
100         echo json_encode(['error' => 'Método no permitido']); exit;
101     }
102
103     $shiftId = (int)$_POST['shift_id'];
104     $assignedShiftIds = $this->institutionShiftModel->getInstitutionShiftIds(1);
105
106     if (in_array($shiftId, $assignedShiftIds)) {
107         $this->institutionShiftModel->remove(1, $shiftId);
108         echo json_encode(['action' => 'removed', 'shift_id' => $shiftId]);
109     } else {
110         $this->institutionShiftModel->assign([
111             'institution_id' => 1,
112             'shift_id'      => $shiftId,
113         ]);
114         echo json_encode(['action' => 'assigned', 'shift_id' => $shiftId]);
115     }
116     exit;
117 }

```

```
118
119 // Mantener compatibilidad con rutas antiguas
120 public function assignShift() {
121     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
122         $this->institutionShiftModel->assign([
123             ':institution_id' => 1,
124             ':shift_id'       => (int)$_POST['shift_id'],
125         ]);
126         header('Location: ?action=institution&success=1');
127         exit;
128     }
129 }
130
131 public function removeShift() {
132     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
133         $this->institutionShiftModel->remove(1, (int)$_POST['shift_id']);
134         header('Location: ?action=institution&success=1');
135         exit;
136     }
137 }
138 }
```

20. ArchImp/controllers/JustificationController.php

```
1  <?php
2  // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3  // controllers/JustificationController.php
4
5  require_once BASE_PATH . '/config/config.php';
6  require_once BASE_PATH . '/models/Justification.php';
7  require_once BASE_PATH . '/models/Attendance.php';
8  require_once BASE_PATH . '/models/Attendance.php';
9  require_once BASE_PATH . '/models/Notification.php';
10
11 class JustificationController {
12     private $justificationModel;
13     private $attendanceModel;
14     private $notificationModel;
15
16     public function __construct() {
17         Security::requireLogin();
18
19         $db = new Database();
20         $this->justificationModel = new Justification($db);
21         $this->attendanceModel = new Attendance($db);
22         $this->notificationModel = new Notification($db);
23     }
24
25     public function submit() {
26         if (!Security::hasRole(['estudiante', 'representante'])) {
27             die('Acceso denegado');
28         }
29
30         $studentId = Security::hasRole('estudiante')
31             ? $_SESSION['user_id']
32             : (isset($_GET['student_id']) ? (int)$_GET['student_id'] : null);
33
34         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
35
36             $studentId = Security::hasRole('estudiante')
37                 ? $_SESSION['user_id']
38                 : (int)$_POST['student_id'];
39
40             // Ausencias seleccionadas
41             $attendanceIds = $_POST['attendance_ids'] ?? [];
42             if (empty($attendanceIds)) {
43                 header('Location: ?action=submit_justification&error=no_absences');
44                 exit;
45             }
46
47             // Obtener fechas de las ausencias seleccionadas para calcular días laborables
48             $db = new Database();
49             $pdo = $db->connect();
50             $in = implode(',', array_map('intval', $attendanceIds));
51             $rows = $pdo->query(
52                 "SELECT date FROM attendances WHERE id IN ($in) AND student_id = $studentId ORDER BY date"
53             )->fetchAll();
54
55             if (empty($rows)) {
56                 header('Location: ?action=submit_justification&error=no_absences');
57                 exit;
58             }
59         }
60     }
61 }
```

```

59
60     $dates          = array_column($rows, 'date');
61     $dateFrom       = min($dates);
62     $dateTo         = max($dates);
63     $workingDays    = count($attendanceIds); // cada ausencia = 1 día (pueden ser no consecutivos)
64     $canApprove     = Justification::resolveApprover($workingDays);
65
66     // Subir documento
67     $documentPath = null;
68     if (isset($_FILES['document']) && $_FILES['document']['error'] == 0) {
69         $uploadDir = BASE_PATH . '/uploads/justifications/';
70         if (!file_exists($uploadDir)) mkdir($uploadDir, 0755, true);
71         $ext       = strtolower(pathinfo($_FILES['document']['name'], PATHINFO_EXTENSION));
72         $allowed   = ['pdf', 'jpg', 'jpeg', 'png'];
73         if (in_array($ext, $allowed) && $_FILES['document']['size'] <= 5 * 1024 * 1024) {
74             $filename = uniqid() . '.' . $ext;
75             if (move_uploaded_file($_FILES['document']['tmp_name'], $uploadDir . $filename)) {
76                 $documentPath = 'uploads/justifications/' . $filename;
77             }
78         }
79     }
80
81     // Motivo
82     $reasonType = Security::sanitize($_POST['reason_type'] ?? '');
83     $reasonText = Security::sanitize($_POST['reason'] ?? '');
84     $reasonFinal = ($reasonType === 'Otro')
85         ? $reasonText
86         : ($reasonType . ($reasonText ? ': ' . $reasonText : ''));
87
88     $data = [
89         ':student_id'    => $studentId,
90         ':submitted_by' => $_SESSION['user_id'],
91         ':date_from'     => $dateFrom,
92         ':date_to'       => $dateTo,
93         ':working_days'  => $workingDays,
94         ':reason_type'   => $reasonType,
95         ':reason'        => $reasonFinal,
96         ':document_path' => $documentPath,
97         ':can_approve'   => $canApprove,
98     ];
99
100    $this->justificationModel->createForAttendances($attendanceIds, $data);
101
102    // Notificar
103    if ($canApprove === 'tutor') {
104        $stmt = $pdo->prepare(
105            "SELECT ta.teacher_id FROM teacher_assignments ta
106            INNER JOIN course_students cs ON ta.course_id = cs.course_id
107            WHERE cs.student_id = :sid AND ta.is_tutor = 1 LIMIT 1"
108        );
109        $stmt->execute([':sid' => $studentId]);
110        $tutorId = $stmt->fetchColumn();
111        if ($tutorId) {
112            $this->notificationModel->create(
113                $tutorId,
114                '■ Justificación pendiente (tutor)',
115                "Un estudiante de tu curso necesita justificación por $workingDays día(s).",
116                'info',
117                '?action=tutor_pending_justifications'

```

```

118         );
119     }
120     } else {
121         $this->_notifyReviewers(
122             '■ Nueva justificación pendiente',
123             "Justificación de $workingDays día(s) requiere revisión.",
124             'info',
125             '?action=pending_justifications'
126         );
127     }
128
129     header('Location: ?action=my_justifications&success=1');
130     exit;
131 }
132
133 // GET: cargar ausencias no justificadas
134 $absences = $this->attendanceModel->getUnjustifiedAbsences($studentId ?? $_SESSION['user_id']);
135
136 // Si representante, cargar sus hijos
137 $myChildren = [];
138 if (Security::hasRole('representante')) {
139     $db = new Database();
140     $stmt = $db->connect()->prepare(
141         "SELECT u.id, CONCAT(u.last_name, ' ', u.first_name) as full_name
142         FROM users u INNER JOIN representatives r ON u.id = r.student_id
143         WHERE r.representative_id = :rid ORDER BY u.last_name"
144     );
145     $stmt->execute([':rid' => $_SESSION['user_id']]);
146     $myChildren = $stmt->fetchAll();
147 }
148
149 include BASE_PATH . '/views/justifications/submit.php';
150 }
151
152
153 public function pendingForTutor() {
154     if (!Security::hasRole('docente')) die('Acceso denegado');
155
156     $db = new Database();
157     $attModel = new Attendance($db);
158     $courseId = $attModel->getTutorCourseId($_SESSION['user_id']);
159
160     if (!$courseId) {
161         header('Location: ?action=dashboard&error=not_tutor'); exit;
162     }
163
164     $justifications = $this->justificationModel->getPendingForTutor($courseId);
165     include BASE_PATH . '/views/justifications/pending.php';
166 }
167
168 public function myJustifications() {
169     if (!Security::hasRole(['estudiante', 'representante'])) {
170         die('Acceso denegado');
171     }
172     $studentId = $_SESSION['user_id'];
173     $justifications = $this->justificationModel->getByStudent($studentId);
174     include BASE_PATH . '/views/justifications/my_list.php';
175 }
176

```

[illegible]


```

236             '■ Justificación rechazada',
237             'La justificación que enviaste fue rechazada.' . ($notes ? " Motivo: $notes" : ''),
238             'danger',
239             '?action=my_justifications'
240         );
241     }
242 }
243 }
244
245     header('Location: ?action=pending_justifications&success=1');
246     exit;
247 }
248 }
249
250 public function reviewed() {
251     if (!Security::hasRole(['autoridad', 'inspector'])) {
252         die('Acceso denegado');
253     }
254     $filter = $_GET['filter'] ?? 'all';
255     $justifications = $this->justificationModel->getReviewed();
256
257     if ($filter !== 'all') {
258         $justifications = array_values(array_filter($justifications, function($j) use ($filter) {
259             return $j['status'] === $filter;
260         }));
261     }
262
263     include BASE_PATH . '/views/justifications/reviewed.php';
264 }
265
266 // Privado: notificar a todos los revisores
267 private function _notifyReviewers($title, $message, $type = 'info', $link = null) {
268     $db = new Database();
269     $pdo = $db->connect();
270
271     $sql = "SELECT DISTINCT u.id FROM users u
272             INNER JOIN user_roles ur ON u.id = ur.user_id
273             INNER JOIN roles r ON ur.role_id = r.id
274             WHERE r.name IN ('autoridad','inspector') AND u.id != :sender";
275     $stmt = $pdo->prepare($sql);
276     $stmt->execute([':sender' => $_SESSION['user_id']]);
277     $reviewers = $stmt->fetchAll(PDO::FETCH_COLUMN);
278
279     foreach ($reviewers as $reviewerId) {
280         $this->notificationModel->create($reviewerId, $title, $message, $type, $link);
281     }
282 }
283 }

```

21. ArchImp/controllers/NotificationController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Notification.php';
4
5  class NotificationController {
6      private $notificationModel;
7
8      public function __construct() {
9          Security::requireLogin();
10         $db = new Database();
11         $this->notificationModel = new Notification($db);
12     }
13
14     // Vista principal: todas las notificaciones del usuario
15     public function index() {
16         $userId = $_SESSION['user_id'];
17         $perPage = 15;
18         $page = max(1, (int)($_GET['page'] ?? 1));
19         $offset = ($page - 1) * $perPage;
20
21         $notifications = $this->notificationModel->getByUser($userId, $perPage, $offset);
22         $total = $this->notificationModel->countByUser($userId);
23         $totalPages = (int)ceil($total / $perPage);
24         $unreadCount = $this->notificationModel->getUnreadCount($userId);
25
26         include BASE_PATH . '/views/notifications/index.php';
27     }
28
29     // AJAX: marcar una como leída y devolver nuevo conteo
30     public function markRead() {
31         $userId = $_SESSION['user_id'];
32         $id = (int)($_GET['id'] ?? 0);
33
34         if ($id) {
35             $this->notificationModel->markAsRead($id, $userId);
36         }
37
38         // Responder JSON para el dropdown del navbar
39         header('Content-Type: application/json');
40         echo json_encode([
41             'success' => true,
42             'unread' => $this->notificationModel->getUnreadCount($userId)
43         ]);
44         exit;
45     }
46
47     // Marcar todas como leídas (POST desde vista)
48     public function markAllRead() {
49         $userId = $_SESSION['user_id'];
50         $this->notificationModel->markAllAsRead($userId);
51         header('Location: ?action=notifications&success=read');
52         exit;
53     }
54
55     // Eliminar una notificación (POST desde vista)
56     public function delete() {
57         $userId = $_SESSION['user_id'];
58         $id = (int)($_POST['notification_id'] ?? 0);
```

```

59
60     if ($id) {
61         $this->notificationModel->delete($id, $userId);
62     }
63
64     header('Location: ?action=notifications&success=deleted');
65     exit;
66 }
67
68 // Eliminar todas las leídas
69 public function deleteRead() {
70     $userId = $_SESSION['user_id'];
71     $this->notificationModel->deleteRead($userId);
72     header('Location: ?action=notifications&success=cleaned');
73     exit;
74 }
75
76 // AJAX: obtener no leídas para el dropdown del navbar
77 public function getUnread() {
78     $userId = $_SESSION['user_id'];
79     $notifications = $this->notificationModel->getUnread($userId, 8);
80     $count = $this->notificationModel->getUnreadCount($userId);
81
82     header('Content-Type: application/json');
83     echo json_encode([
84         'notifications' => $notifications,
85         'count' => $count
86     ]);
87     exit;
88 }
89 }

```

22. ArchImp/controllers/ProfileController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/User.php';
4
5  class ProfileController {
6      private $userModel;
7
8      public function __construct() {
9          Security::requireLogin();
10
11          $db = new Database();
12          $this->userModel = new User($db);
13      }
14
15      public function view() {
16          $user = $this->userModel->findById($_SESSION['user_id']);
17          $roles = $this->userModel->getUserRoles($_SESSION['user_id']);
18
19          include BASE_PATH . '/views/profile/view.php';
20      }
21
22      public function edit() {
23          $user = $this->userModel->findById($_SESSION['user_id']);
24
25          if ($_SERVER['REQUEST_METHOD'] === 'POST') {
26              $data = [
27                  'first_name' => Security::sanitize($_POST['first_name']),
28                  'last_name' => Security::sanitize($_POST['last_name']),
29                  'phone' => Security::sanitize($_POST['phone']),
30                  'email' => Security::sanitize($_POST['email'])
31              ];
32
33              if ($this->userModel->updateProfile($_SESSION['user_id'], $data)) {
34                  header('Location: ?action=profile&success=1');
35                  exit;
36              }
37          }
38
39          include BASE_PATH . '/views/profile/edit.php';
40      }
41
42      public function changePassword() {
43          if ($_SERVER['REQUEST_METHOD'] === 'POST') {
44              $currentPassword = $_POST['current_password'];
45              $newPassword = $_POST['new_password'];
46              $confirmPassword = $_POST['confirm_password'];
47
48              $user = $this->userModel->findById($_SESSION['user_id']);
49
50              if (!Security::verifyPassword($currentPassword, $user['password'])) {
51                  $error = 'Contraseña actual incorrecta';
52              } elseif ($newPassword !== $confirmPassword) {
53                  $error = 'Las contraseñas no coinciden';
54              } elseif (strlen($newPassword) < 6) {
55                  $error = 'La contraseña debe tener al menos 6 caracteres';
56              } else {
57                  if ($this->userModel->updatePassword($_SESSION['user_id'], $newPassword)) {
58                      header('Location: ?action=profile&password_changed=1');
```

```
59             exit;
60         }
61     }
62 }
63
64     include BASE_PATH . '/views/profile/change_password.php';
65 }
66 }
```

23. ArchImp/controllers/ReportController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Attendance.php';
4  require_once BASE_PATH . '/models/Course.php';
5  require_once BASE_PATH . '/models/Subject.php';
6  require_once BASE_PATH . '/models/SchoolYear.php';
7  require_once BASE_PATH . '/models/User.php';
8  require_once BASE_PATH . '/models/Institution.php';
9  require_once BASE_PATH . '/vendor/autoload.php';
10
11 use PhpOffice\PhpSpreadsheet\Spreadsheet;
12 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
13 use PhpOffice\PhpSpreadsheet\Style\Fill;
14 use PhpOffice\PhpSpreadsheet\Style\Border;
15 use PhpOffice\PhpSpreadsheet\Style\Alignment;
16
17 class ReportController {
18     private $attendanceModel;
19     private $courseModel;
20     private $subjectModel;
21     private $schoolYearModel;
22     private $userModel;
23     private $institutionModel;
24
25     public function __construct() {
26         Security::requireLogin();
27         if (!Security::hasRole(['autoridad', 'inspector', 'docente'])) {
28             die('Acceso denegado');
29         }
30
31         $db = new Database();
32         $this->attendanceModel = new Attendance($db);
33         $this->courseModel = new Course($db);
34         $this->subjectModel = new Subject($db);
35         $this->schoolYearModel = new SchoolYear($db);
36         $this->userModel = new User($db);
37         $this->institutionModel = new Institution($db);
38     }
39
40     public function index() {
41         $courses = $this->courseModel->getAll();
42         $subjects = $this->subjectModel->getAll();
43
44         $data = [];
45         $course = null;
46         $startDate = null;
47         $endDate = null;
48
49         if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['preview'])) {
50             $courseId = (int)$_POST['course_id'];
51             $startDate = $_POST['start_date'];
52             $endDate = $_POST['end_date'];
53
54             $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
55             $course = $this->getCourseInfo($courseId);
56         }
57
58         include BASE_PATH . '/views/reports/index.php';
```

```

59     }
60
61     public function generatePDF() {
62         $courseId = (int)$_POST['course_id'];
63         $startDate = $_POST['start_date'];
64         $endDate = $_POST['end_date'];
65
66         $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
67         $course = $this->getCourseInfo($courseId);
68         $institution = $this->institutionModel->getById($_SESSION['institution_id']);
69
70         $pdf = new \TCPDF('L', 'mm', 'A4', true, 'UTF-8');
71
72         $pdf->SetCreator('EcuAsist 2026');
73         $pdf->SetAuthor('Sistema de Asistencia');
74         $pdf->SetTitle('Reporte de Asistencia');
75
76         $pdf->setPrintHeader(false);
77         $pdf->setPrintFooter(false);
78         $pdf->SetMargins(10, 10, 10);
79         $pdf->SetAutoPageBreak(true, 10);
80
81         $pdf->AddPage();
82
83         $pdf->SetFont('helvetica', 'B', 16);
84         $pdf->Cell(0, 10, 'REPORTE DE ASISTENCIA', 0, 1, 'C');
85
86         $pdf->SetFont('helvetica', '', 10);
87         $pdf->Cell(0, 5, $institution['name'] ?? 'Institución Educativa', 0, 1, 'C');
88         $pdf->Ln(5);
89
90         $pdf->SetFont('helvetica', 'B', 11);
91         $pdf->Cell(50, 6, 'Curso:', 0, 0);
92         $pdf->SetFont('helvetica', '', 11);
93         $pdf->Cell(0, 6, html_entity_decode($course['name'], ENT_QUOTES, 'UTF-8'), 0, 1);
94
95         $pdf->SetFont('helvetica', 'B', 11);
96         $pdf->Cell(50, 6, 'Período:', 0, 0);
97         $pdf->SetFont('helvetica', '', 11);
98         $pdf->Cell(0, 6, date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)), 0, 1);
99         $pdf->Ln(3);
100
101         $html = '<table border="1" cellpadding="4">
102             <thead>
103                 <tr style="background-color: #007bff; color: white; font-weight: bold;">
104                     <th width="16.66666%">Fecha</th>
105                     <th width="16.66666%">Estudiante</th>
106                     <th width="16.66666%">Asignatura</th>
107                     <th width="16.66666%">Hora</th>
108                     <th width="16.66666%">Estado</th>
109                     <th width="16.66666%">Observación</th>
110                 </tr>
111             </thead>
112             <tbody>';
113
114         foreach ($data as $row) {
115             $statusColor = $this->getStatusColor($row['status']);
116             $html .= '<tr>
117                 <td>' . date('d/m/Y', strtotime($row['date'])) . '</td>';

```

```

118         &lt;td&gt;' . $row['student_name'] . '&lt;/td&gt;
119         &lt;td&gt;' . $row['subject_name'] . '&lt;/td&gt;
120         &lt;td&gt;' . $row['hour_period'] . '&lt;/td&gt;
121         &lt;td style="background-color: ' . $statusColor . ';"&gt;' . ucfirst($row['status']) . '&lt;/td.
122         &lt;td&gt;' . ($row['observation'] ?: '-') . '&lt;/td&gt;
123     &lt;/tr&gt;';
124 }
125
126 $html .= '&lt;/tbody&gt;&lt;/table&gt;';
127
128 $pdf-&gt;SetFont('helvetica','',$9);
129
130 $pdf-&gt;writeHTML($html, true, false, true, false, '');
131
132 // Generar nombre de archivo con curso
133 $courseName = $this-&gt;sanitizeFilename($course['name']);
134 $filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.pdf';
135
136 $pdf-&gt;Output($filename, 'D');
137 exit;
138 }
139
140 public function generateExcel() {
141     $courseId = (int)$_POST['course_id'];
142     $startDate = $_POST['start_date'];
143     $endDate = $_POST['end_date'];
144
145     $data = $this-&gt;attendanceModel-&gt;getReportData($courseId, $startDate, $endDate);
146     $course = $this-&gt;getCourseInfo($courseId);
147     $institution = $this-&gt;institutionModel-&gt;getById($_SESSION['institution_id']);
148
149     $spreadsheet = new Spreadsheet();
150     $sheet = $spreadsheet-&gt;getActiveSheet();
151
152     $sheet-&gt;setTitle('Asistencia');
153
154     // Encabezado
155     $sheet-&gt;mergeCells('A1:F1');
156     $sheet-&gt;setCellValue('A1', 'REPORTE DE ASISTENCIA');
157     $sheet-&gt;getStyle('A1')-&gt;getFont()-&gt;setBold(true)-&gt;setSize(16);
158     $sheet-&gt;getStyle('A1')-&gt;getAlignment()-&gt;setHorizontal(Alignment::HORIZONTAL_CENTER);
159
160     $sheet-&gt;mergeCells('A2:F2');
161     $sheet-&gt;setCellValue('A2', $institution['name'] ?? 'Institución Educativa');
162     $sheet-&gt;getStyle('A2')-&gt;getAlignment()-&gt;setHorizontal(Alignment::HORIZONTAL_CENTER);
163
164     $sheet-&gt;setCellValue('A4', 'Curso:');
165     $sheet-&gt;setCellValue('B4', html_entity_decode($course['name'], ENT_QUOTES, 'UTF-8'));
166     $sheet-&gt;getStyle('A4')-&gt;getFont()-&gt;setBold(true);
167
168     $sheet-&gt;setCellValue('A5', 'Período:');
169     $sheet-&gt;setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)));
170     $sheet-&gt;getStyle('A5')-&gt;getFont()-&gt;setBold(true);
171
172     // Cabecera de tabla
173     $row = 7;
174     $headers = ['Fecha', 'Estudiante', 'Asignatura', 'Hora', 'Estado', 'Observación'];
175     $col = 'A';
176

```



```

177     foreach ($headers as $header) {
178         $sheet->setCellValue($col . $row, $header);
179         $sheet->getStyle($col . $row)->getFont()->setBold(true);
180         $sheet->getStyle($col . $row)->getFill()
181             ->setFillType(Fill::FILL_SOLID)
182             ->getStartColor()->setRGB('007bff');
183         $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
184         $col++;
185     }
186
187     // Datos
188     $row = 8;
189     foreach ($data as $record) {
190         $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
191         $sheet->setCellValue('B' . $row, $record['student_name']);
192         $sheet->setCellValue('C' . $row, $record['subject_name']);
193         $sheet->setCellValue('D' . $row, $record['hour_period']);
194         $sheet->setCellValue('E' . $row, ucfirst($record['status']));
195         $sheet->setCellValue('F' . $row, $record['observation'] ?: '-');
196
197         $statusColor = $this->getStatusColorHex($record['status']);
198         $sheet->getStyle('E' . $row)->getFill()
199             ->setFillType(Fill::FILL_SOLID)
200             ->getStartColor()->setRGB($statusColor);
201
202         $row++;
203     }
204
205     // Ajustar anchos
206     $sheet->getColumnDimension('A')->setWidth(12);
207     $sheet->getColumnDimension('B')->setWidth(25);
208     $sheet->getColumnDimension('C')->setWidth(20);
209     $sheet->getColumnDimension('D')->setWidth(15);
210     $sheet->getColumnDimension('E')->setWidth(12);
211     $sheet->getColumnDimension('F')->setWidth(25);
212
213     // Bordes
214     $styleArray = [
215         'borders' => [
216             'allBorders' => [
217                 'borderStyle' => Border::BORDER_THIN,
218             ],
219         ],
220     ];
221     $sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);
222
223     $writer = new Xlsx($spreadsheet);
224
225     // Generar nombre de archivo con curso
226     $courseName = $this->sanitizeFilename($course['name']);
227     $filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.xlsx';
228
229     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
230     header('Content-Disposition: attachment;filename="' . $filename . '"');
231     header('Cache-Control: max-age=0');
232
233     $writer->save('php://output');
234     exit;
235 }

```

```

236
237     private function getCourseInfo($courseId) {
238         $courses = $this->courseModel->getAll();
239         foreach ($courses as $course) {
240             if ($course['id'] == $courseId) {
241                 return $course;
242             }
243         }
244         return null;
245     }
246
247     private function getStatusColor($status) {
248         switch ($status) {
249             case 'presente': return '#d4edda';
250             case 'ausente': return '#f8d7da';
251             case 'tardanza': return '#fff3cd';
252             case 'justificado': return '#d1ecf1';
253             default: return '#ffffff';
254         }
255     }
256
257     private function getStatusColorHex($status) {
258         switch ($status) {
259             case 'presente': return '28a745';
260             case 'ausente': return 'dc3545';
261             case 'tardanza': return 'ffc107';
262             case 'justificado': return '17a2b8';
263             default: return 'FFFFFF';
264         }
265     }
266
267     private function sanitizeFilename($filename) {
268         // Decodificar HTML entities
269         $filename = html_entity_decode($filename, ENT_QUOTES, 'UTF-8');
270
271         // Reemplazar caracteres no permitidos en nombres de archivo
272         $filename = str_replace(['"', '/', '\\', ':', '*', '?', '<', '>', '|'], '', $filename);
273
274         // Reemplazar espacios con guiones bajos
275         $filename = str_replace(' ', '_', $filename);
276
277         // Reemplazar guiones múltiples
278         $filename = preg_replace('/-/+', '-', $filename);
279
280         // Limitar longitud (máximo 50 caracteres)
281         if (strlen($filename) > 50) {
282             $filename = substr($filename, 0, 50);
283         }
284
285         // Limpiar caracteres al inicio y final
286         $filename = trim($filename, '-_');
287
288         return $filename;
289     }
290 }

```

24. ArchImp/controllers/ReportController2.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/vendor/autoload.php';
4  require_once BASE_PATH . '/models/Attendance.php';
5  require_once BASE_PATH . '/models/Course.php';
6  require_once BASE_PATH . '/models/Subject.php';
7  require_once BASE_PATH . '/models/SchoolYear.php';
8  require_once BASE_PATH . '/models/User.php';
9
10 use PhpOffice\PhpSpreadsheet\Spreadsheet;
11 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
12 use PhpOffice\PhpSpreadsheet\Style\Fill;
13 use PhpOffice\PhpSpreadsheet\Style\Border;
14 use PhpOffice\PhpSpreadsheet\Style\Alignment;
15
16 class ReportController {
17     private $attendanceModel;
18     private $courseModel;
19     private $subjectModel;
20     private $schoolYearModel;
21     private $userModel;
22
23     public function __construct() {
24         Security::requireLogin();
25         if (!Security::hasRole(['autoridad', 'inspector', 'docente'])) {
26             die('Acceso denegado');
27         }
28
29         $db = new Database();
30         $this->attendanceModel = new Attendance($db);
31         $this->courseModel = new Course($db);
32         $this->subjectModel = new Subject($db);
33         $this->schoolYearModel = new SchoolYear($db);
34         $this->userModel = new User($db);
35     }
36
37     public function index() {
38         $courses = $this->courseModel->getAll();
39         $subjects = $this->subjectModel->getAll();
40
41         include BASE_PATH . '/views/reports/index.php';
42     }
43
44     public function generatePDF() {
45         $courseId = (int)$_POST['course_id'];
46         $startDate = $_POST['start_date'];
47         $endDate = $_POST['end_date'];
48
49         $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
50         $course = $this->getCourseInfo($courseId);
51
52         try {
53             $pdf = new \TCPDF('L', 'mm', 'A4', true, 'UTF-8');
54
55             $pdf->SetCreator('EcuAsist 2026');
56             $pdf->SetAuthor('Sistema de Asistencia');
57             $pdf->SetTitle('Reporte de Asistencia');
58         }
```

```

59     $pdf->setPrintHeader(false);
60     $pdf->setPrintFooter(false);
61     $pdf->setMargins(10, 10, 10);
62     $pdf->setAutoPageBreak(true, 10);
63
64     $pdf->AddPage();
65
66     $pdf->setFont('helvetica', 'B', 16);
67     $pdf->Cell(0, 10, 'REPORTE DE ASISTENCIA', 0, 1, 'C');
68
69     $pdf->setFont('helvetica', '', 10);
70     $pdf->Cell(0, 5, 'Unidad Educativa Demo', 0, 1, 'C');
71     $pdf->Ln(5);
72
73     $pdf->setFont('helvetica', 'B', 11);
74     $pdf->Cell(50, 6, 'Curso:', 0, 0);
75     $pdf->setFont('helvetica', '', 11);
76     $pdf->Cell(0, 6, $course['name'] . ' - ' . $course['shift_name'], 0, 1);
77
78     $pdf->setFont('helvetica', 'B', 11);
79     $pdf->Cell(50, 6, 'Período:', 0, 0);
80     $pdf->setFont('helvetica', '', 11);
81     $pdf->Cell(0, 6, date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)), 0, 1);
82     $pdf->Ln(3);
83
84     $html = '<table border="1" cellpadding="4">';
85         <thead>
86             <tr style="background-color: #007bff; color: white; font-weight: bold;">
87                 <th width="10%">Fecha</th>
88                 <th width="25%">Estudiante</th>
89                 <th width="20%">Asignatura</th>
90                 <th width="15%">Hora</th>
91                 <th width="12%">Estado</th>
92                 <th width="18%">Observación</th>
93             </tr>
94         </thead>
95         <tbody>
96
97         foreach ($data as $row) {
98             $statusColor = $this->getStatusColor($row['status']);
99             $html .= '<tr>
100                 <td>' . date('d/m/Y', strtotime($row['date'])) . '</td>
101                 <td>' . htmlspecialchars($row['student_name']) . '</td>
102                 <td>' . htmlspecialchars($row['subject_name']) . '</td>
103                 <td>' . htmlspecialchars($row['hour_period']) . '</td>
104                 <td style="background-color: ' . $statusColor . ';">' . ucfirst($row['status']) . '&
105                 <td>' . htmlspecialchars($row['observation'] ?: '-') . '</td>
106             </tr>';
107         }
108
109     $html .= '</tbody></table>';
110
111     $pdf->writeHTML($html, true, false, true, false, '');
112
113     $pdf->Output('reporte_asistencia_' . date('YmdHis') . '.pdf', 'D');
114     exit;
115
116 } catch (Exception $e) {
117     die('Error generando PDF: ' . $e->getMessage());

```

```

118     }
119 }
120
121 public function generateExcel() {
122     $courseId = (int)$_POST['course_id'];
123     $startDate = $_POST['start_date'];
124     $endDate = $_POST['end_date'];
125
126     $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
127     $course = $this->getCourseInfo($courseId);
128
129     try {
130         $spreadsheet = new Spreadsheet();
131         $sheet = $spreadsheet->getActiveSheet();
132
133         $sheet->setTitle('Asistencia');
134
135         // Encabezado
136         $sheet->mergeCells('A1:F1');
137         $sheet->setCellValue('A1', 'REPORTE DE ASISTENCIA');
138         $sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
139         $sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);
140
141         $sheet->mergeCells('A2:F2');
142         $sheet->setCellValue('A2', 'Unidad Educativa Demo');
143         $sheet->getStyle('A2')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);
144
145         $sheet->setCellValue('A4', 'Curso:');
146         $sheet->setCellValue('B4', $course['name'] . ' - ' . $course['shift_name']);
147         $sheet->getStyle('A4')->getFont()->setBold(true);
148
149         $sheet->setCellValue('A5', 'Período:');
150         $sheet->setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)));
151         $sheet->getStyle('A5')->getFont()->setBold(true);
152
153         // Cabecera de tabla
154         $row = 7;
155         $headers = ['Fecha', 'Estudiante', 'Asignatura', 'Hora', 'Estado', 'Observación'];
156         $col = 'A';
157
158         foreach ($headers as $header) {
159             $sheet->setCellValue($col . $row, $header);
160             $sheet->getStyle($col . $row)->getFont()->setBold(true);
161             $sheet->getStyle($col . $row)->getFill()
162                 ->setFillType(Fill::FILL_SOLID)
163                 ->getStartColor()->setRGB('007bff');
164             $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
165             $col++;
166         }
167
168         // Datos
169         $row = 8;
170         foreach ($data as $record) {
171             $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
172             $sheet->setCellValue('B' . $row, $record['student_name']);
173             $sheet->setCellValue('C' . $row, $record['subject_name']);
174             $sheet->setCellValue('D' . $row, $record['hour_period']);
175             $sheet->setCellValue('E' . $row, ucfirst($record['status']));
176             $sheet->setCellValue('F' . $row, $record['observation'] ? '-' : '');

```

```

177
178         $statusColor = $this->getStatusColorHex($record['status']);
179         $sheet->getStyle('E' . $row)->getFill()
180             ->setFillType(Fill::FILL_SOLID)
181             ->getStartColor()->setRGB($statusColor);
182
183         $row++;
184     }
185
186     // Ajustar anchos
187     $sheet->getColumnDimension('A')->setWidth(12);
188     $sheet->getColumnDimension('B')->setWidth(25);
189     $sheet->getColumnDimension('C')->setWidth(20);
190     $sheet->getColumnDimension('D')->setWidth(15);
191     $sheet->getColumnDimension('E')->setWidth(12);
192     $sheet->getColumnDimension('F')->setWidth(25);
193
194     // Bordes
195     $styleArray = [
196         'borders' => [
197             'allBorders' => [
198                 'borderStyle' => Border::BORDER_THIN,
199             ],
200         ],
201     ];
202     $sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);
203
204     $writer = new Xlsx($spreadsheet);
205
206     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
207     header('Content-Disposition: attachment;filename="reporte_asistencia_' . date('YmdHis') . '.xlsx"');
208     header('Cache-Control: max-age=0');
209
210     $writer->save('php://output');
211     exit;
212
213     } catch (Exception $e) {
214         die('Error generando Excel: ' . $e->getMessage());
215     }
216 }
217
218 private function getCourseInfo($courseId) {
219     $courses = $this->courseModel->getAll();
220     foreach ($courses as $course) {
221         if ($course['id'] == $courseId) {
222             return $course;
223         }
224     }
225     return null;
226 }
227
228 private function getStatusColor($status) {
229     switch ($status) {
230         case 'presente': return '#d4edda';
231         case 'ausente': return '#f8d7da';
232         case 'tardanza': return '#fff3cd';
233         case 'justificado': return '#d1ecf1';
234         default: return '#ffffff';
235     }

```

```
236     }
237
238     private function getStatusColorHex($status) {
239         switch ($status) {
240             case 'presente': return 'd4edda';
241             case 'ausente': return 'f8d7da';
242             case 'tardanza': return 'fff3cd';
243             case 'justificado': return 'd1ecf1';
244             default: return 'FFFFFF';
245         }
246     }
247 }
```

25. ArchImp/controllers/RepresentativeController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Representative.php';
4  require_once BASE_PATH . '/models/User.php';
5  require_once BASE_PATH . '/models/Attendance.php';
6
7  class RepresentativeController {
8      private $representativeModel;
9      private $userModel;
10     private $attendanceModel;
11
12     public function __construct() {
13         Security::requireLogin();
14
15         $db = new Database();
16         $this->representativeModel = new Representative($db);
17         $this->userModel = new User($db);
18         $this->attendanceModel = new Attendance($db);
19     }
20
21     public function manageRepresentatives() {
22         if (!Security::hasRole('autoridad')) {
23             die('Acceso denegado');
24         }
25
26         $representatives = $this->userModel->getByRole('representante');
27         $students = $this->userModel->getByRole('estudiante');
28
29         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
30             $repId = (int)$_POST['representative_id'];
31             $studentId = (int)$_POST['student_id'];
32             $relationship = Security::sanitize($_POST['relationship']);
33             $isPrimary = isset($_POST['is_primary']) ? 1 : 0;
34
35             $this->representativeModel->assignStudent($repId, $studentId, $relationship, $isPrimary);
36             header('Location: ?action=manage_representatives&success=1');
37             exit;
38         }
39
40         include BASE_PATH . '/views/representatives/manage.php';
41     }
42
43     public function myChildren() {
44         if (!Security::hasRole('representante')) {
45             die('Acceso denegado');
46         }
47
48         $children = $this->representativeModel->getStudentsByRepresentative($_SESSION['user_id']);
49
50         include BASE_PATH . '/views/representatives/my_children.php';
51     }
52
53     public function childAttendance() {
54         if (!Security::hasRole('representante')) {
55             die('Acceso denegado');
56         }
57
58         $studentId = (int)($_GET['student_id'] ?? 0);
```



```

59
60     // Verificar que el estudiante sea hijo del representante
61     $children = $this->representativeModel->getStudentsByRepresentative($_SESSION['user_id']);
62     $authorized = false;
63     $student = null;
64
65     foreach ($children as $child) {
66         if ($child['id'] == $studentId) {
67             $authorized = true;
68             $student = $child;
69             break;
70         }
71     }
72
73     if (!$authorized) {
74         die('No autorizado para ver este estudiante');
75     }
76
77     $attendances = $this->attendanceModel->getByStudent($studentId);
78
79     include BASE_PATH . '/views/representatives/child_attendance.php';
80 }
81
82 public function removeRelation() {
83     if (!Security::hasRole('autoridad')) {
84         die('Acceso denegado');
85     }
86
87     $repId = (int)($_GET['rep_id'] ?? 0);
88     $studentId = (int)($_GET['student_id'] ?? 0);
89
90     if ($repId && $studentId) {
91         if ($this->representativeModel->removeStudent($repId, $studentId)) {
92             header('Location: ?action=manage_representatives&removed=1');
93             exit;
94         }
95     }
96
97     header('Location: ?action=manage_representatives&error=1');
98     exit;
99 }
100 }

```

26. ArchImp/controllers/ScheduleController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/ClassSchedule.php';
4  require_once BASE_PATH . '/models/Course.php';
5  require_once BASE_PATH . '/models/Subject.php';
6  require_once BASE_PATH . '/models/User.php';
7  require_once BASE_PATH . '/models/SchoolYear.php';
8
9  class ScheduleController {
10     private $scheduleModel;
11     private $courseModel;
12     private $subjectModel;
13     private $userModel;
14     private $schoolYearModel;
15
16     public function __construct() {
17         Security::requireLogin();
18         if (!Security::hasRole('autoridad')) {
19             die('Acceso denegado');
20         }
21
22         $db = new Database();
23         $this->scheduleModel = new ClassSchedule($db);
24         $this->courseModel = new Course($db);
25         $this->subjectModel = new Subject($db);
26         $this->userModel = new User($db);
27         $this->schoolYearModel = new SchoolYear($db);
28     }
29
30     public function index() {
31         $courses = $this->courseModel->getAll();
32         include BASE_PATH . '/views/schedule/index.php';
33     }
34
35     public function manageCourse() {
36         $courseId = (int)($_GET['course_id'] ?? 0);
37
38         if (!$courseId) {
39             header('Location: ?action=schedules');
40             exit;
41         }
42
43         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
44             $activeYear = $this->schoolYearModel->getActive();
45
46             $data = [
47                 ':course_id' => $courseId,
48                 ':subject_id' => (int)$_POST['subject_id'],
49                 ':teacher_id' => (int)$_POST['teacher_id'],
50                 ':school_year_id' => $activeYear['id'],
51                 ':day_of_week' => $_POST['day_of_week'],
52                 ':period_number' => (int)$_POST['period_number']
53             ];
54
55             $result = $this->scheduleModel->create($data);
56
57             if ($result['success']) {
58                 header('Location: ?action=manage_schedule&course_id=' . $courseId . '&success=1');
```

```

59         } else {
60             header('Location: ?action=manage_schedule&course_id=' . $courseId . '&error=' . urlencode($result));
61         }
62         exit;
63     }
64
65     $activeYear = $this->schoolYearModel->getActive();
66     $courses = $this->courseModel->getAll();
67     $course = array_filter($courses, fn($c) => $c['id'] == $courseId);
68     $course = reset($course);
69
70     $schedule = $this->scheduleModel->getByCourse($courseId, $activeYear['id']);
71     $subjects = $this->subjectModel->getAll();
72     $teachers = $this->userModel->getByRole('docente');
73
74     include BASE_PATH . '/views/schedule/manage.php';
75 }
76
77 public function deleteClass() {
78     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
79         $scheduleId = (int)$_POST['schedule_id'];
80         $courseId = (int)$_POST['course_id'];
81
82         $this->scheduleModel->delete($scheduleId);
83         header('Location: ?action=manage_schedule&course_id=' . $courseId . '&deleted=1');
84         exit;
85     }
86 }
87
88 public function getCourseSubjectsSchedule() {
89     if (!isset($_SESSION['user_id'])) {
90         header('Content-Type: application/json');
91         echo json_encode([]);
92         exit;
93     }
94
95     $courseId = (int)($_GET['course_id'] ?? 0);
96
97     if (!$courseId) {
98         header('Content-Type: application/json');
99         echo json_encode([]);
100         exit;
101     }
102
103     // Obtener asignaturas con sus docentes asignados al curso
104     $sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name,
105            ta.teacher_id, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
106            FROM teacher_assignments ta
107            INNER JOIN subjects s ON ta.subject_id = s.id
108            INNER JOIN users u ON ta.teacher_id = u.id
109            WHERE ta.course_id = :course_id
110            ORDER BY s.name";
111
112     $db = new Database();
113     $stmt = $db->connect()->prepare($sql);
114     $stmt->execute([':course_id' => $courseId]);
115     $subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);
116
117     header('Content-Type: application/json');

```

```

118         echo json_encode($subjects);
119         exit;
120     }
121
122     public function checkScheduleConflict() {
123         if (!isset($_SESSION['user_id'])) {
124             header('Content-Type: application/json');
125             echo json_encode(['exists' => false]);
126             exit;
127         }
128
129         $courseId = (int)($_GET['course_id'] ?? 0);
130         $day = $_GET['day'] ?? '';
131         $period = (int)($_GET['period'] ?? 0);
132
133         if (!$courseId || !$day || !$period) {
134             header('Content-Type: application/json');
135             echo json_encode(['exists' => false]);
136             exit;
137         }
138
139         $db = new Database();
140         $activeYear = $this->schoolYearModel->getActive();
141
142         $sql = "SELECT s.name as subject_name, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
143             FROM class_schedule cs
144             INNER JOIN subjects s ON cs.subject_id = s.id
145             INNER JOIN users u ON cs.teacher_id = u.id
146             WHERE cs.course_id = :course_id
147             AND cs.day_of_week = :day
148             AND cs.period_number = :period
149             AND cs.school_year_id = :school_year_id
150             LIMIT 1";
151
152         $stmt = $db->connect()->prepare($sql);
153         $stmt->execute([
154             ':course_id' => $courseId,
155             ':day' => $day,
156             ':period' => $period,
157             ':school_year_id' => $activeYear['id']
158         ]);
159
160         $existing = $stmt->fetch(PDO::FETCH_ASSOC);
161
162         $result = [
163             'exists' => $existing ? true : false,
164             'subject_name' => $existing['subject_name'] ?? null,
165             'teacher_name' => $existing['teacher_name'] ?? null
166         ];
167
168         header('Content-Type: application/json');
169         echo json_encode($result);
170         exit;
171     }
172 }

```

27. ArchImp/controllers/SearchController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/User.php';
4  require_once BASE_PATH . '/models/Course.php';
5  require_once BASE_PATH . '/models/Attendance.php';
6
7  class SearchController {
8      private $userModel;
9      private $courseModel;
10     private $attendanceModel;
11
12     public function __construct() {
13         Security::requireLogin();
14
15         $db = new Database();
16         $this->userModel = new User($db);
17         $this->courseModel = new Course($db);
18         $this->attendanceModel = new Attendance($db);
19     }
20
21     public function searchStudents() {
22         $query = $_GET['q'] ?? '';
23
24         if (strlen($query) < 2) {
25             echo json_encode([]);
26             exit;
27         }
28
29         $sql = "SELECT u.id, u.first_name, u.last_name, u.dni, u.email,
30             c.name as course_name
31             FROM users u
32             INNER JOIN user_roles ur ON u.id = ur.user_id
33             INNER JOIN roles r ON ur.role_id = r.id
34             LEFT JOIN course_students cs ON u.id = cs.student_id
35             LEFT JOIN courses c ON cs.course_id = c.id
36             WHERE r.name = 'estudiante'
37             AND u.institution_id = :institution_id
38             AND (u.first_name LIKE :query
39                 OR u.last_name LIKE :query
40                 OR u.dni LIKE :query
41                 OR u.email LIKE :query)
42             LIMIT 10";
43
44         $db = new Database();
45         $stmt = $db->connect()->prepare($sql);
46         $stmt->execute([
47             ':institution_id' => $_SESSION['institution_id'],
48             ':query' => '%' . $query . '%'
49         ]);
50
51         header('Content-Type: application/json');
52         echo json_encode($stmt->fetchAll());
53         exit;
54     }
55
56     public function advancedAttendanceReport() {
57         if (!Security::hasRole(['autoridad', 'inspector'])) {
58             die('Acceso denegado');
```

```
59     }
60
61     $filters = [
62         'course_id' => $_POST['course_id'] ?? null,
63         'student_id' => $_POST['student_id'] ?? null,
64         'subject_id' => $_POST['subject_id'] ?? null,
65         'status' => $_POST['status'] ?? null,
66         'start_date' => $_POST['start_date'] ?? null,
67         'end_date' => $_POST['end_date'] ?? null
68     ];
69
70     $results = $this->attendanceModel->getFiltered($filters);
71
72     include BASE_PATH . '/views/reports/advanced.php';
73 }
74 }
```

28. ArchImp/controllers/StatsController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/Attendance.php';
4  require_once BASE_PATH . '/models/Course.php';
5  require_once BASE_PATH . '/models/User.php';
6
7  class StatsController {
8      private $attendanceModel;
9      private $courseModel;
10     private $userModel;
11
12     public function __construct() {
13         Security::requireLogin();
14         if (!Security::hasRole('autoridad')) {
15             die('Acceso denegado');
16         }
17
18         $db = new Database();
19         $this->attendanceModel = new Attendance($db);
20         $this->courseModel = new Course($db);
21         $this->userModel = new User($db);
22     }
23
24     public function index() {
25         $startDate = $_GET['start_date'] ?? date('Y-m-01');
26         $endDate = $_GET['end_date'] ?? date('Y-m-d');
27
28         $stats = [
29             'overall' => $this->getOverallStats($startDate, $endDate),
30             'by_course' => $this->getStatsByCourse($startDate, $endDate),
31             'by_student' => $this->getTopAbsentStudents($startDate, $endDate),
32             'by_day' => $this->getStatsByDay($startDate, $endDate)
33         ];
34
35         include BASE_PATH . '/views/stats/index.php';
36     }
37
38     private function getOverallStats($startDate, $endDate) {
39         $sql = "SELECT
40             COUNT(*) as total_records,
41             COUNT(DISTINCT student_id) as total_students,
42             SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
43             SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
44             SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,
45             SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
46             FROM attendances
47             WHERE date BETWEEN :start_date AND :end_date";
48
49         $db = new Database();
50         $stmt = $db->connect()->prepare($sql);
51         $stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
52         return $stmt->fetch();
53     }
54
55     private function getStatsByCourse($startDate, $endDate) {
56         $sql = "SELECT
57             c.name as course_name,
58             COUNT(*) as total,
```

```

59         SUM(CASE WHEN a.status = 'presente' THEN 1 ELSE 0 END) as presente,
60         SUM(CASE WHEN a.status = 'ausente' THEN 1 ELSE 0 END) as ausente,
61         ROUND(SUM(CASE WHEN a.status = 'presente' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as percentage
62     FROM attendances a
63     INNER JOIN courses c ON a.course_id = c.id
64     WHERE a.date BETWEEN :start_date AND :end_date
65     GROUP BY c.id, c.name
66     ORDER BY percentage DESC";
67
68     $db = new Database();
69     $stmt = $db->connect()->prepare($sql);
70     $stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
71     return $stmt->fetchAll();
72 }
73
74 private function getTopAbsentStudents($startDate, $endDate, $limit = 10) {
75     $sql = "SELECT
76         CONCAT(u.last_name, ' ', u.first_name) as student_name,
77         c.name as course_name,
78         COUNT(*) as total_absences
79     FROM attendances a
80     INNER JOIN users u ON a.student_id = u.id
81     INNER JOIN courses c ON a.course_id = c.id
82     WHERE a.status = 'ausente'
83     AND a.date BETWEEN :start_date AND :end_date
84     GROUP BY a.student_id, u.last_name, u.first_name, c.name
85     ORDER BY total_absences DESC
86     LIMIT :limit";
87
88     $db = new Database();
89     $stmt = $db->connect()->prepare($sql);
90     $stmt->bindValue(':start_date', $startDate);
91     $stmt->bindValue(':end_date', $endDate);
92     $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
93     $stmt->execute();
94     return $stmt->fetchAll();
95 }
96
97 private function getStatsByDay($startDate, $endDate) {
98     $sql = "SELECT
99         date,
100         COUNT(*) as total,
101         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
102         ROUND(SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as percentage
103     FROM attendances
104     WHERE date BETWEEN :start_date AND :end_date
105     GROUP BY date
106     ORDER BY date ASC";
107
108     $db = new Database();
109     $stmt = $db->connect()->prepare($sql);
110     $stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
111     return $stmt->fetchAll();
112 }
113 }

```


29. ArchImp/controllers/TutorController.php

```
1  <?php
2  // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3  // controllers/TutorController.php
4
5  require_once BASE_PATH . '/config/config.php';
6  require_once BASE_PATH . '/models/Attendance.php';
7  require_once BASE_PATH . '/models/Course.php';
8
9  class TutorController {
10     private $attendanceModel;
11     private $courseModel;
12
13     public function __construct() {
14         Security::requireLogin();
15         if (!Security::hasRole('docente')) {
16             die('Acceso denegado');
17         }
18         $db = new Database();
19         $this->attendanceModel = new Attendance($db);
20         $this->courseModel = new Course($db);
21     }
22
23     public function courseAttendance() {
24         $teacherId = $_SESSION['user_id'];
25         $courseId = $this->attendanceModel->getTutorCourseId($teacherId);
26
27         if (!$courseId) {
28             include BASE_PATH . '/views/tutor/no_tutor.php';
29             return;
30         }
31
32         $course = $this->courseModel->findById($courseId);
33         $subjects = $this->attendanceModel->getSubjectsByCourse($courseId);
34         $students = $this->attendanceModel->getStudentsByCourse($courseId);
35         $topAbsences = $this->attendanceModel->getTutorTopAbsences($courseId, 5);
36
37         // Leer filtros del GET
38         $filters = [];
39         if (!empty($_GET['subject_id'])) $filters['subject_id'] = (int)$_GET['subject_id'];
40         if (!empty($_GET['student_id'])) $filters['student_id'] = (int)$_GET['student_id'];
41         if (!empty($_GET['status'])) $filters['status'] = $_GET['status'];
42         if (!empty($_GET['start_date'])) $filters['start_date'] = $_GET['start_date'];
43         if (!empty($_GET['end_date'])) $filters['end_date'] = $_GET['end_date'];
44
45         $attendances = $this->attendanceModel->getTutorCourseAttendance($courseId, $filters);
46         $stats = $this->attendanceModel->getTutorCourseStats($courseId, $filters);
47
48         include BASE_PATH . '/views/tutor/course_attendance.php';
49     }
50
51     // Endpoint AJAX
52     public function ajax() {
53         header('Content-Type: application/json');
54
55         $teacherId = $_SESSION['user_id'];
56         $courseId = $this->attendanceModel->getTutorCourseId($teacherId);
57
58         if (!$courseId) {
```

[illegible]

[illegible]

30. ArchImp/controllers/UserController.php

```
1  <?php
2  require_once BASE_PATH . '/config/config.php';
3  require_once BASE_PATH . '/models/User.php';
4  require_once BASE_PATH . '/models/Role.php';
5
6  class UserController {
7      private $userModel;
8      private $roleModel;
9
10     public function __construct() {
11         Security::requireLogin();
12         if (!Security::hasRole('autoridad')) {
13             die('Acceso denegado');
14         }
15
16         $db = new Database();
17         $this->userModel = new User($db);
18         $this->roleModel = new Role($db);
19     }
20
21     public function index() {
22         $users = $this->userModel->getAll();
23         $roles = $this->roleModel->getAll();
24         include BASE_PATH . '/views/users/index.php';
25     }
26
27     public function assignRole() {
28         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
29             $userId = (int)$_POST['user_id'];
30             $roleId = (int)$_POST['role_id'];
31
32             $this->userModel->assignRole($userId, $roleId);
33             $filter = isset($_GET['filter_role']) ? '&filter_role=' . $_GET['filter_role'] : '';
34             header('Location: ?action=users&success=1' . $filter);
35             exit;
36         }
37     }
38
39     public function removeRole() {
40         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
41             $userId = (int)$_POST['user_id'];
42             $roleId = (int)$_POST['role_id'];
43
44             // Verificar si el rol es "docente"
45             $db = new Database();
46             $stmt = $db->connect()->prepare("SELECT name FROM roles WHERE id = :role_id");
47             $stmt->execute([':role_id' => $roleId]);
48             $role = $stmt->fetch();
49
50             if ($role && $role['name'] === 'docente') {
51                 // Eliminar asignaciones de materias
52                 $stmt = $db->connect()->prepare("DELETE FROM teacher_assignments WHERE teacher_id = :user_id");
53                 $stmt->execute([':user_id' => $userId]);
54             }
55
56             $this->userModel->removeRole($userId, $roleId);
57             $filter = isset($_GET['filter_role']) ? '&filter_role=' . $_GET['filter_role'] : '';
58             header('Location: ?action=users&removed=1' . $filter);
```

```

59         exit;
60     }
61 }
62
63 public function create() {
64     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
65         $roles = $this->roleModel->getAll();
66         include BASE_PATH . '/views/users/create.php';
67         return;
68     }
69
70     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
71         $errors = [];
72
73         // Validaciones
74         if (empty($_POST['username'])) {
75             $errors[] = "El nombre de usuario es obligatorio";
76         }
77         if (empty($_POST['email']) || !filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
78             $errors[] = "Email inválido";
79         }
80         if (empty($_POST['password']) || strlen($_POST['password']) < 6) {
81             $errors[] = "La contraseña debe tener al menos 6 caracteres";
82         }
83         if ($_POST['password'] !== $_POST['confirm_password']) {
84             $errors[] = "Las contraseñas no coinciden";
85         }
86         if (empty($_POST['first_name'])) {
87             $errors[] = "El nombre es obligatorio";
88         }
89         if (empty($_POST['last_name'])) {
90             $errors[] = "El apellido es obligatorio";
91         }
92
93         // Verificar username único
94         if ($this->userModel->findByUsername($_POST['username'])) {
95             $errors[] = "El nombre de usuario ya está registrado";
96         }
97
98         // Verificar email único
99         if ($this->userModel->findByEmail($_POST['email'])) {
100             $errors[] = "El email ya está registrado";
101         }
102
103         // Validar cédula ecuatoriana (si no es extranjero y hay valor)
104         $esExtranjero = !empty($_POST['es_extranjero']);
105         $dniValue = null;
106         if (!$esExtranjero && !empty($_POST['dni'])) {
107             $ced = preg_replace('/\D/', '', $_POST['dni']);
108             if (!$this->validarCedulaEcuador($ced)) {
109                 $errors[] = "Cédula ecuatoriana inválida";
110             } else {
111                 $dniValue = $ced;
112             }
113         } elseif ($esExtranjero && !empty($_POST['passport'])) {
114             $dniValue = Security::sanitize($_POST['passport']);
115         }
116
117         // Validar teléfono Ecuador

```

```

118     $phoneValue = null;
119     if (!empty($_POST['phone'])) {
120         $tel = preg_replace('/[\s-]/', '', $_POST['phone']);
121         if (!preg_match('/^09\d{8}$/', $tel) && !preg_match('/^0[2-7]\d{7}$/', $tel)) {
122             $errors[] = "Teléfono inválido. Celular: 09XXXXXXXX · Fijo: 0XXXXXXXX";
123         } else {
124             $phoneValue = $tel;
125         }
126     }
127
128     if (!empty($errors)) {
129         $roles = $this->roleModel->getAll();
130         include BASE_PATH . '/views/users/create.php';
131         return;
132     }
133
134     // Crear usuario
135     $userData = [
136         'institution_id' => $_SESSION['institution_id'],
137         'username'       => Security::sanitize($_POST['username']),
138         'email'          => Security::sanitize($_POST['email']),
139         'password'       => $_POST['password'],
140         'first_name'     => Security::sanitize($_POST['first_name']),
141         'last_name'      => Security::sanitize($_POST['last_name']),
142         'dni'            => $dniValue,
143         'phone'          => $phoneValue,
144     ];
145
146     if ($this->userModel->create($userData)) {
147         // Obtener ID del usuario recién creado
148         $newUserId = $this->userModel->db->lastInsertId();
149
150         // Asignar roles si se seleccionaron
151         if (!empty($_POST['roles'])) {
152             foreach ($_POST['roles'] as $roleId) {
153                 $this->userModel->assignRole($newUserId, (int)$roleId);
154             }
155         }
156
157         header('Location: ?action=users&created=1');
158         exit;
159     } else {
160         $errors[] = "Error al crear el usuario";
161         $roles = $this->roleModel->getAll();
162         include BASE_PATH . '/views/users/create.php';
163     }
164 }
165
166
167 public function edit() {
168     $userId = (int)$_GET['id'];
169     $user = $this->userModel->findById($userId);
170
171     if (!$user) {
172         header('Location: ?action=users&error=not_found');
173         exit;
174     }
175
176     // Verificar que pertenece a la misma institución

```

```

177     if ($user['institution_id'] != $_SESSION['institution_id']) {
178         die('Acceso denegado');
179     }
180
181     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
182         $roles = $this->roleModel->getAll();
183         $userRoles = $this->userModel->getUserRoleIds($userId);
184         include BASE_PATH . '/views/users/edit.php';
185         return;
186     }
187
188     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
189         $errors = [];
190
191         // Validaciones
192         if (empty($_POST['username'])) {
193             $errors[] = "El nombre de usuario es obligatorio";
194         }
195         if (empty($_POST['email']) || !filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
196             $errors[] = "Email inválido";
197         }
198         if (empty($_POST['first_name'])) {
199             $errors[] = "El nombre es obligatorio";
200         }
201         if (empty($_POST['last_name'])) {
202             $errors[] = "El apellido es obligatorio";
203         }
204
205         // Verificar username único (excepto el actual)
206         $existingUser = $this->userModel->findByUsername($_POST['username']);
207         if ($existingUser && $existingUser['id'] != $userId) {
208             $errors[] = "El nombre de usuario ya está registrado";
209         }
210
211         // Verificar email único (excepto el actual)
212         $existingEmail = $this->userModel->findByEmail($_POST['email']);
213         if ($existingEmail && $existingEmail['id'] != $userId) {
214             $errors[] = "El email ya está registrado";
215         }
216
217         // Si hay nueva contraseña, validarla
218         if (!empty($_POST['password'])) {
219             if (strlen($_POST['password']) < 6) {
220                 $errors[] = "La contraseña debe tener al menos 6 caracteres";
221             }
222             if ($_POST['password'] !== $_POST['confirm_password']) {
223                 $errors[] = "Las contraseñas no coinciden";
224             }
225         }
226
227         if (!empty($errors)) {
228             $roles = $this->roleModel->getAll();
229             $userRoles = $this->userModel->getUserRoleIds($userId);
230             include BASE_PATH . '/views/users/edit.php';
231             return;
232         }
233
234         // Actualizar usuario
235         $updateData = [

```



```

236         'id' => $userId,
237         'username' => Security::sanitize($_POST['username']),
238         'email' => Security::sanitize($_POST['email']),
239         'first_name' => Security::sanitize($_POST['first_name']),
240         'last_name' => Security::sanitize($_POST['last_name']),
241         'dni' => !empty($_POST['dni']) ? Security::sanitize($_POST['dni']) : null,
242         'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null
243     ];
244
245     // Solo actualizar contraseña si se proporcionó una nueva
246     if (!empty($_POST['password'])) {
247         $updateData['password'] = $_POST['password'];
248     }
249
250     if ($this->userModel->update($updateData)) {
251         header('Location: ?action=users&updated=1');
252         exit;
253     } else {
254         $errors[] = "Error al actualizar el usuario";
255         $roles = $this->roleModel->getAll();
256         $userRoles = $this->userModel->getUserRoleIds($userId);
257         include BASE_PATH . '/views/users/edit.php';
258     }
259 }
260 }
261
262 public function delete() {
263     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
264         $userId = (int)$_POST['user_id'];
265         $user = $this->userModel->findById($userId);
266
267         if (!$user) {
268             header('Location: ?action=users&error=not_found');
269             exit;
270         }
271
272         // Verificar que pertenece a la misma institución
273         if ($user['institution_id'] !== $_SESSION['institution_id']) {
274             die('Acceso denegado');
275         }
276
277         // No permitir eliminar al propio usuario
278         if ($userId == $_SESSION['user_id']) {
279             header('Location: ?action=users&error=self_delete');
280             exit;
281         }
282
283         // Verificar si tiene asistencias registradas
284         $db = new Database();
285         $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM attendances WHERE student_id = :user_id.");
286         $stmt->execute([':user_id' => $userId]);
287         $result = $stmt->fetch();
288
289         if ($result['count'] > 0) {
290             // Soft delete - desactivar en lugar de eliminar
291             if ($this->userModel->deactivate($userId)) {
292                 header('Location: ?action=users&deactivated=1');
293                 exit;
294             }

```

[illegible]

31. ArchImp/helpers/Backup.php

```
1  <?php
2  class Backup {
3      private $dbHost = 'localhost';
4      private $dbUser = 'root';
5      private $dbPass = '';
6      private $dbName = 'ecuasistencia2026_db';
7      private $backupDir;
8
9      public function __construct() {
10         $this->backupDir = BASE_PATH . '/backups/';
11         if (!file_exists($this->backupDir)) {
12             mkdir($this->backupDir, 0755, true);
13         }
14     }
15
16     public function createBackup() {
17         $filename = 'backup_' . date('Y-m-d_H-i-s') . '.sql';
18         $filepath = $this->backupDir . $filename;
19
20         // Verificar que mysqldump esté disponible
21         $mysqldumpPath = 'C:\\xampp\\mysql\\bin\\mysqldump.exe'; // XAMPP Windows
22         if (!file_exists($mysqldumpPath)) {
23             $mysqldumpPath = 'mysqldump'; // Linux/Mac o PATH configurado
24         }
25
26         // Comando mejorado con opciones
27         if (empty($this->dbPass)) {
28             $command = sprintf(
29                 "%s --host=%s --user=%s --skip-password %s &gt; \"%s\" 2&gt;&amp;1",
30                 $mysqldumpPath,
31                 $this->dbHost,
32                 $this->dbUser,
33                 $this->dbName,
34                 $filepath
35             );
36         } else {
37             $command = sprintf(
38                 "%s --host=%s --user=%s --password=%s %s &gt; \"%s\" 2&gt;&amp;1",
39                 $mysqldumpPath,
40                 $this->dbHost,
41                 $this->dbUser,
42                 $this->dbPass,
43                 $this->dbName,
44                 $filepath
45             );
46         }
47
48         exec($command, $output, $result);
49
50         // Verificar que el archivo existe y tiene contenido
51         if (file_exists($filepath) && filesize($filepath) &gt; 0) {
52             return $filename;
53         }
54
55         // Si falló, intentar eliminar archivo vacío
56         if (file_exists($filepath)) {
57             unlink($filepath);
58         }
59     }
60 }
```

```

59
60     return false;
61 }
62
63 public function getBackups() {
64     $files = glob($this->backupDir . 'backup_*.sql');
65     $backups = [];
66
67     foreach ($files as $file) {
68         $size = filesize($file);
69         $backups[] = [
70             'filename' => basename($file),
71             'path' => $file,
72             'size' => $this->formatBytes($size),
73             'date' => date('d/m/Y H:i:s', filemtime($file)),
74             'timestamp' => filemtime($file)
75         ];
76     }
77
78     usort($backups, function($a, $b) {
79         return $b['timestamp'] - $a['timestamp'];
80     });
81
82     return $backups;
83 }
84
85 private function formatBytes($bytes, $precision = 2) {
86     $units = ['B', 'KB', 'MB', 'GB'];
87     $bytes = max($bytes, 0);
88     $pow = floor(($bytes ? log($bytes) : 0) / log(1024));
89     $pow = min($pow, count($units) - 1);
90     $bytes /= (1 << ($pow * 10));
91     return round($bytes, $precision) . ' ' . $units[$pow];
92 }
93
94 public function deleteOldBackups($daysToKeep = 30) {
95     $files = glob($this->backupDir . 'backup_*.sql');
96     $deleted = 0;
97
98     foreach ($files as $file) {
99         if (time() - filemtime($file) > ($daysToKeep * 24 * 60 * 60)) {
100             unlink($file);
101             $deleted++;
102         }
103     }
104
105     return $deleted;
106 }
107
108 public function downloadBackup($filename) {
109     $filepath = $this->backupDir . $filename;
110
111     if (!file_exists($filepath)) {
112         return false;
113     }
114
115     header('Content-Type: application/octet-stream');
116     header('Content-Disposition: attachment; filename="' . $filename . '"');
117     header('Content-Length: ' . filesize($filepath));

```

```
118         readfile($filepath);
119         exit;
120     }
121
122     public function deleteBackup($filename) {
123         // Validar que sea un archivo de backup válido
124         if (!preg_match('/^backup_[\d\-\_]+\\.sql$/', $filename)) {
125             return false;
126         }
127
128         $filepath = $this->backupDir . $filename;
129
130         if (file_exists($filepath)) {
131             return unlink($filepath);
132         }
133
134         return false;
135     }
136 }
```

32. ArchImp/helpers/Logger.php

```
1  <?php
2  class Logger {
3      private $db;
4
5      public function __construct() {
6          $database = new Database();
7          $this->db = $database->connect();
8      }
9
10     public function log($action, $entityType = null, $entityId = null, $description = null) {
11         if (!isset($_SESSION['user_id'])) {
12             return false;
13         }
14
15         $sql = "INSERT INTO activity_logs
16             (user_id, action, entity_type, entity_id, description, ip_address, user_agent)
17             VALUES (:user_id, :action, :entity_type, :entity_id, :description, :ip_address, :user_agent)";
18
19         $stmt = $this->db->prepare($sql);
20         return $stmt->execute([
21             ':user_id' => $_SESSION['user_id'],
22             ':action' => $action,
23             ':entity_type' => $entityType,
24             ':entity_id' => $entityId,
25             ':description' => $description,
26             ':ip_address' => $_SERVER['REMOTE_ADDR'] ?? null,
27             ':user_agent' => $_SERVER['HTTP_USER_AGENT'] ?? null
28         ]);
29     }
30
31     public function getLogs($limit = 100, $filters = []) {
32         $sql = "SELECT l.*, CONCAT(u.last_name, ' ', u.first_name) as user_name
33             FROM activity_logs l
34             INNER JOIN users u ON l.user_id = u.id
35             WHERE 1=1";
36
37         $params = [];
38
39         if (!empty($filters['user_id'])) {
40             $sql .= " AND l.user_id = :user_id";
41             $params[':user_id'] = $filters['user_id'];
42         }
43
44         if (!empty($filters['action'])) {
45             $sql .= " AND l.action = :action";
46             $params[':action'] = $filters['action'];
47         }
48
49         if (!empty($filters['start_date'])) {
50             $sql .= " AND l.created_at >= :start_date";
51             $params[':start_date'] = $filters['start_date'];
52         }
53
54         if (!empty($filters['end_date'])) {
55             $sql .= " AND l.created_at <= :end_date";
56             $params[':end_date'] = $filters['end_date'] . ' 23:59:59';
57         }
58     }
```

```
59     $sql .= " ORDER BY l.created_at DESC LIMIT :limit";
60
61     $stmt = $this->db->prepare($sql);
62     foreach ($params as $key => $value) {
63         $stmt->bindValue($key, $value);
64     }
65     $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
66     $stmt->execute();
67
68     return $stmt->fetchAll();
69 }
70 }
```

33. ArchImp/helpers/Mailer.php

```
1  <?php
2  use PHPMailer\PHPMailer\PHPMailer;
3  use PHPMailer\PHPMailer\Exception;
4
5  require BASE_PATH . '/vendor/autoload.php';
6
7  class Mailer {
8      public static function send($to, $subject, $body) {
9          $mail = new PHPMailer(true);
10
11          try {
12              $mail->isSMTP();
13              $mail->Host = SMTP_HOST;
14              $mail->SMTPAuth = true;
15              $mail->Username = SMTP_USER;
16              $mail->Password = SMTP_PASS;
17              $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
18              $mail->Port = SMTP_PORT;
19              $mail->CharSet = 'UTF-8';
20
21              $mail->setFrom(SMTP_FROM, SMTP_NAME);
22              $mail->addAddress($to);
23
24              $mail->isHTML(true);
25              $mail->Subject = $subject;
26              $mail->Body = $body;
27
28              $mail->send();
29              return true;
30          } catch (Exception $e) {
31              error_log("Error de correo: {$mail->ErrorInfo}");
32              return false;
33          }
34      }
35  }
```


34. ArchImp/helpers/Security.php

```
1  <?php
2  class Security {
3      public static function generateToken() {
4          if (!isset($_SESSION['csrf_token'])) {
5              $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
6          }
7          return $_SESSION['csrf_token'];
8      }
9
10     public static function validateToken($token) {
11         return isset($_SESSION['csrf_token']) && hash_equals($_SESSION['csrf_token'], $token);
12     }
13
14     public static function sanitize($data) {
15         return htmlspecialchars(strip_tags(trim($data)), ENT_QUOTES, 'UTF-8');
16     }
17
18     public static function validateEmail($email) {
19         return filter_var($email, FILTER_VALIDATE_EMAIL);
20     }
21
22     public static function hashPassword($password) {
23         return password_hash($password, PASSWORD_BCRYPT);
24     }
25
26     public static function verifyPassword($password, $hash) {
27         return password_verify($password, $hash);
28     }
29
30     public static function requireLogin() {
31         if (!isset($_SESSION['user_id'])) {
32             header('Location: ' . BASE_URL . '/public/index.php?action=login');
33             exit;
34         }
35     }
36
37     public static function hasRole($roleNames) {
38         // Superusuario siempre tiene acceso
39         if (!empty($_SESSION['is_superuser'])) return true;
40         if (!isset($_SESSION['roles'])) return false;
41         $roleNames = (array) $roleNames;
42         return count(array_intersect($roleNames, $_SESSION['roles'])) > 0;
43     }
44
45     public static function isSuperAdmin() {
46         return !empty($_SESSION['is_superuser']);
47     }
48
49     public static function startSession($rememberMe = false) {
50         if ($rememberMe) {
51             // Sesión de 7 días
52             $lifetime = 7 * 24 * 60 * 60; // 7 días
53         } else {
54             // Sesión expira al cerrar navegador
55             $lifetime = 0;
56         }
57
58         session_set_cookie_params([
```

```
59         'lifetime' => $lifetime,
60         'path' => '/',
61         'httponly' => true,
62         'samesite' => 'Lax'
63     ];
64
65     session_start();
66
67     // Regenerar ID por seguridad
68     if (!isset($_SESSION['initiated'])) {
69         session_regenerate_id(true);
70         $_SESSION['initiated'] = true;
71     }
72
73     // Timeout de inactividad (30 minutos)
74     if (isset($_SESSION['last_activity']) &&
75         (time() - $_SESSION['last_activity'] > 1800)) {
76         session_unset();
77         session_destroy();
78         return false;
79     }
80
81     $_SESSION['last_activity'] = time();
82     return true;
83 }
84 }
```

35. ArchImp/models/Attendance.php

```
1  <?php
2  // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3  // models/Attendance.php
4
5  class Attendance {
6      private $db;
7
8      public function __construct($database) {
9          $this->db = $database->connect();
10     }
11
12     public function create($data) {
13         $checkSql = "SELECT id FROM attendances
14                     WHERE student_id = :student_id
15                     AND course_id = :course_id
16                     AND subject_id = :subject_id
17                     AND date = :date
18                     AND hour_period = :hour_period";
19         $checkStmt = $this->db->prepare($checkSql);
20         $checkStmt->execute([
21             ':student_id' => $data[':student_id'],
22             ':course_id'  => $data[':course_id'],
23             ':subject_id' => $data[':subject_id'],
24             ':date'       => $data[':date'],
25             ':hour_period' => $data[':hour_period']
26         ]);
27         $existing = $checkStmt->fetch();
28
29         if ($existing) {
30             $updateSql = "UPDATE attendances
31                         SET status = :status, observation = :observation, updated_at = NOW()
32                         WHERE id = :id";
33             $updateStmt = $this->db->prepare($updateSql);
34             return $updateStmt->execute([
35                 ':status'      => $data[':status'],
36                 ':observation' => $data[':observation'],
37                 ':id'          => $existing['id']
38             ]);
39         }
40
41         $sql = "INSERT INTO attendances
42              (student_id, course_id, subject_id, teacher_id, school_year_id, shift_id, date, hour_period, status, obs.
43              VALUES (:student_id, :course_id, :subject_id, :teacher_id, :school_year_id, :shift_id, :date, :hour_peri.
44         $stmt = $this->db->prepare($sql);
45         return $stmt->execute($data);
46     }
47
48     public function canEdit($attendanceId) {
49         $sql = "SELECT created_at FROM attendances WHERE id = :id";
50         $stmt = $this->db->prepare($sql);
51         $stmt->execute([':id' => $attendanceId]);
52         $record = $stmt->fetch();
53         if (!$record) return false;
54         return (time() - strtotime($record['created_at'])) <= (EDIT_ATTENDANCE_HOURS * 3600);
55     }
56
57     public function update($id, $data) {
58         $sql = "UPDATE attendances
```

```

59         SET status = :status, observation = :observation, updated_at = NOW()
60         WHERE id = :id";
61     $stmt = $this->db->prepare($sql);
62     return $stmt->execute([':id' => $id, ':status' => $data['status'], ':observation' => $data['observat.
63 }
64
65 public function getByCourse($courseId, $date) {
66     $sql = "SELECT a.*, u.first_name, u.last_name, s.name as subject_name,
67         CONCAT(t.first_name, ' ', t.last_name) as teacher_name
68     FROM attendances a
69     INNER JOIN users u ON a.student_id = u.id
70     INNER JOIN subjects s ON a.subject_id = s.id
71     INNER JOIN users t ON a.teacher_id = t.id
72     WHERE a.course_id = :course_id AND a.date = :date
73     ORDER BY u.last_name, u.first_name";
74     $stmt = $this->db->prepare($sql);
75     $stmt->execute([':course_id' => $courseId, ':date' => $date]);
76     return $stmt->fetchAll();
77 }
78
79 public function getByStudent($studentId) {
80     $sql = "SELECT a.*, s.name as subject_name, c.name as course_name, sh.name as shift_name
81     FROM attendances a
82     INNER JOIN subjects s ON a.subject_id = s.id
83     INNER JOIN courses c ON a.course_id = c.id
84     INNER JOIN shifts sh ON a.shift_id = sh.id
85     WHERE a.student_id = :student_id
86     ORDER BY a.date DESC, a.hour_period
87     LIMIT 100";
88     $stmt = $this->db->prepare($sql);
89     $stmt->execute([':student_id' => $studentId]);
90     return $stmt->fetchAll();
91 }
92
93 public function getReportData($courseId, $startDate, $endDate) {
94     $sql = "SELECT a.*,
95         CONCAT(u.last_name, ' ', u.first_name) as student_name,
96         s.name as subject_name
97     FROM attendances a
98     INNER JOIN users u ON a.student_id = u.id
99     INNER JOIN subjects s ON a.subject_id = s.id
100     WHERE a.course_id = :course_id
101     AND a.date BETWEEN :start_date AND :end_date
102     ORDER BY a.date, u.last_name, u.first_name";
103     $stmt = $this->db->prepare($sql);
104     $stmt->execute([':course_id' => $courseId, ':start_date' => $startDate, ':end_date' => $endDate]);
105     return $stmt->fetchAll();
106 }
107
108 public function getStatsByCourse($courseId, $startDate, $endDate) {
109     $sql = "SELECT
110         COUNT(*) as total,
111         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
112         SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
113         SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,
114         SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
115     FROM attendances
116     WHERE course_id = :course_id AND date BETWEEN :start_date AND :end_date";
117     $stmt = $this->db->prepare($sql);

```

[illegible]

```

177         WHERE ta.course_id = :course_id
178         ORDER BY s.name";
179     $stmt = $this->db->prepare($sql);
180     $stmt->execute([':course_id' => $courseId]);
181     return $stmt->fetchAll();
182 }
183
184 // Estudiantes del curso (para filtro)
185 public function getStudentsByCourse($courseId) {
186     $sql = "SELECT u.id, CONCAT(u.last_name, ' ', u.first_name) as full_name
187         FROM users u
188         INNER JOIN course_students cs ON u.id = cs.student_id
189         WHERE cs.course_id = :course_id
190         ORDER BY u.last_name, u.first_name";
191     $stmt = $this->db->prepare($sql);
192     $stmt->execute([':course_id' => $courseId]);
193     return $stmt->fetchAll();
194 }
195
196 // Asistencias del curso con filtros (vista del tutor)
197 public function getTutorCourseAttendance($courseId, $filters = []) {
198     $sql = "SELECT
199         a.id, a.date, a.hour_period, a.status, a.observation,
200         CONCAT(u.last_name, ' ', u.first_name) as student_name,
201         u.id as student_id,
202         s.name as subject_name,
203         s.id as subject_id,
204         CONCAT(t.last_name, ' ', t.first_name) as teacher_name
205     FROM attendances a
206     INNER JOIN users u ON a.student_id = u.id
207     INNER JOIN subjects s ON a.subject_id = s.id
208     INNER JOIN users t ON a.teacher_id = t.id
209     WHERE a.course_id = :course_id";
210     $params = [':course_id' => $courseId];
211
212     if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params[':subject_id'] = $filt.
213     if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params[':student_id'] = $filt.
214     if (!empty($filters['status'])) { $sql .= " AND a.status = :status"; $params[':status'] = $filt.
215     if (!empty($filters['start_date'])) { $sql .= " AND a.date >= :start_date"; $params[':start_date'] = $f.
216     if (!empty($filters['end_date'])) { $sql .= " AND a.date <= :end_date"; $params[':end_date'] = $f.
217
218     $sql .= " ORDER BY a.date DESC, u.last_name, u.first_name, s.name";
219     $stmt = $this->db->prepare($sql);
220     $stmt->execute($params);
221     return $stmt->fetchAll();
222 }
223
224 // Estadísticas globales del curso (vista del tutor)
225 public function getTutorCourseStats($courseId, $filters = []) {
226     $sql = "SELECT
227         COUNT(*) as total,
228         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
229         SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
230         SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,
231         SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
232     FROM attendances a
233     WHERE a.course_id = :course_id";
234     $params = [':course_id' => $courseId];
235

```

```

236         if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params[':subject_id'] = $f.
237         if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params[':student_id'] = $f.
238         if (!empty($filters['start_date'])) { $sql .= " AND a.date >= :start_date"; $params[':start_date'] = $f.
239         if (!empty($filters['end_date'])) { $sql .= " AND a.date <= :end_date"; $params[':end_date'] = $f.
240
241         $stmt = $this->db->prepare($sql);
242         $stmt->execute($params);
243         return $stmt->fetch();
244     }
245
246     // Top estudiantes con más ausencias (vista del tutor)
247     public function getTutorTopAbsences($courseId, $limit = 10) {
248         $sql = "SELECT CONCAT(u.last_name, ' ', u.first_name) as student_name,
249             COUNT(*) as total_ausencias
250             FROM attendances a
251             INNER JOIN users u ON a.student_id = u.id
252             WHERE a.course_id = :course_id AND a.status = 'ausente'
253             GROUP BY a.student_id, u.last_name, u.first_name
254             ORDER BY total_ausencias DESC
255             LIMIT :lim";
256         $stmt = $this->db->prepare($sql);
257         $stmt->bindValue(':course_id', $courseId, PDO::PARAM_INT);
258         $stmt->bindValue(':lim', $limit, PDO::PARAM_INT);
259         $stmt->execute();
260         return $stmt->fetchAll();
261     }
262
263     // Ausencias sin justificación pendiente o aprobada (para el formulario de justificación)
264     public function getUnjustifiedAbsences($studentId) {
265         $sql = "SELECT a.id, a.date, a.hour_period, a.status,
266             s.name as subject_name,
267             c.name as course_name
268             FROM attendances a
269             INNER JOIN subjects s ON a.subject_id = s.id
270             INNER JOIN courses c ON a.course_id = c.id
271             WHERE a.student_id = :student_id
272             AND a.status = 'ausente'
273             AND a.id NOT IN (
274                 SELECT attendance_id FROM justifications
275                 WHERE attendance_id IS NOT NULL
276                 AND status IN ('pendiente','aprobado')
277             )
278             ORDER BY a.date DESC, a.hour_period ASC";
279         $stmt = $this->db->prepare($sql);
280         $stmt->execute([':student_id' => $studentId]);
281         return $stmt->fetchAll();
282     }
283 }

```

36. ArchImp/models/ClassSchedule.php

```
1  <?php
2  class ClassSchedule {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function create($data) {
10         // Verificar si ya existe una clase en ese horario
11         $checkSql = "SELECT s.name as subject_name,
12                     CONCAT(u.last_name, ' ', u.first_name) as teacher_name
13                     FROM class_schedule cs
14                     INNER JOIN subjects s ON cs.subject_id = s.id
15                     INNER JOIN users u ON cs.teacher_id = u.id
16                     WHERE cs.course_id = :course_id
17                     AND cs.day_of_week = :day_of_week
18                     AND cs.period_number = :period_number
19                     AND cs.school_year_id = :school_year_id";
20
21         $checkStmt = $this->db->prepare($checkSql);
22         $checkStmt->execute([
23             ':course_id' => $data[':course_id'],
24             ':day_of_week' => $data[':day_of_week'],
25             ':period_number' => $data[':period_number'],
26             ':school_year_id' => $data[':school_year_id']
27         ]);
28
29         $existing = $checkStmt->fetch();
30
31         if ($existing) {
32             return [
33                 'success' => false,
34                 'message' => 'Ya existe ' . $existing['subject_name'] . ' con ' . $existing['teacher_name'] . ' en es.
35             ];
36         }
37
38         $sql = "INSERT INTO class_schedule
39             (course_id, subject_id, teacher_id, school_year_id, day_of_week, period_number)
40             VALUES (:course_id, :subject_id, :teacher_id, :school_year_id, :day_of_week, :period_number)";
41
42         $stmt = $this->db->prepare($sql);
43         $result = $stmt->execute($data);
44
45         return [
46             'success' => $result,
47             'message' => 'Clase agregada correctamente'
48         ];
49     }
50
51     public function getByCourse($courseId, $schoolYearId) {
52         $sql = "SELECT cs.*,
53                 s.name as subject_name,
54                 CONCAT(u.last_name, ' ', u.first_name) as teacher_name
55                 FROM class_schedule cs
56                 INNER JOIN subjects s ON cs.subject_id = s.id
57                 INNER JOIN users u ON cs.teacher_id = u.id
58                 WHERE cs.course_id = :course_id
```



```

59         AND cs.school_year_id = :school_year_id
60         ORDER BY cs.day_of_week, cs.period_number";
61
62     $stmt = $this->db->prepare($sql);
63     $stmt->execute([
64         ':course_id' => $courseId,
65         ':school_year_id' => $schoolYearId
66     ]);
67     return $stmt->fetchAll();
68 }
69
70 public function getTeacherScheduleToday($teacherId, $schoolYearId) {
71     $dayName = $this->getCurrentDayName();
72
73     $sql = "SELECT cs.*,
74             c.name as course_name,
75             c.grade_level,
76             s.name as subject_name
77     FROM class_schedule cs
78     INNER JOIN courses c ON cs.course_id = c.id
79     INNER JOIN subjects s ON cs.subject_id = s.id
80     WHERE cs.teacher_id = :teacher_id
81     AND cs.school_year_id = :school_year_id
82     AND cs.day_of_week = :day_of_week
83     ORDER BY cs.period_number";
84
85     $stmt = $this->db->prepare($sql);
86     $stmt->execute([
87         ':teacher_id' => $teacherId,
88         ':school_year_id' => $schoolYearId,
89         ':day_of_week' => $dayName
90     ]);
91     return $stmt->fetchAll();
92 }
93
94 public function delete($id) {
95     $sql = "DELETE FROM class_schedule WHERE id = :id";
96     $stmt = $this->db->prepare($sql);
97     return $stmt->execute([':id' => $id]);
98 }
99
100 private function getCurrentDayName() {
101     // Forzar zona horaria Ecuador
102     $tz = new DateTimeZone('America/Guayaquil');
103     $now = new DateTime('now', $tz);
104     $dayNumber = (int)$now->format('N'); // 1=Lunes, 7=Domingo
105
106     $days = [
107         1 => 'lunes',
108         2 => 'martes',
109         3 => 'miercoles',
110         4 => 'jueves',
111         5 => 'viernes',
112         6 => 'sabado'
113     ];
114
115     return $days[$dayNumber] ?? 'lunes';
116 }
117 }

```


37. ArchImp/models/Course.php

```
1  <?php
2  class Course {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getAll() {
10         $sql = "SELECT c.*, s.name as shift_name, sy.name as year_name
11             FROM courses c
12             INNER JOIN shifts s ON c.shift_id = s.id
13             INNER JOIN school_years sy ON c.school_year_id = sy.id
14             WHERE c.institution_id = :institution_id
15             ORDER BY c.name";
16
17         $stmt = $this->db->prepare($sql);
18         $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
19         return $stmt->fetchAll();
20     }
21
22     public function create($data) {
23         $sql = "INSERT INTO courses (institution_id, school_year_id, name, grade_level, parallel, shift_id)
24             VALUES (:institution_id, :school_year_id, :name, :grade_level, :parallel, :shift_id)";
25
26         $stmt = $this->db->prepare($sql);
27         return $stmt->execute($data);
28     }
29
30     public function getStudents($courseId) {
31         $sql = "SELECT u.id, u.first_name, u.last_name, u.dni
32             FROM users u
33             INNER JOIN course_students cs ON u.id = cs.student_id
34             WHERE cs.course_id = :course_id
35             ORDER BY u.last_name, u.first_name";
36
37         $stmt = $this->db->prepare($sql);
38         $stmt->execute([':course_id' => $courseId]);
39         return $stmt->fetchAll();
40     }
41
42     public function enrollStudent($courseId, $studentId, $schoolYearId) {
43         // Primero verificar si ya está matriculado en otro curso del mismo año
44         $checkSql = "SELECT COUNT(*) as count FROM course_students
45             WHERE student_id = :student_id AND school_year_id = :school_year_id";
46         $checkStmt = $this->db->prepare($checkSql);
47         $checkStmt->execute([
48             ':student_id' => $studentId,
49             ':school_year_id' => $schoolYearId
50         ]);
51         $result = $checkStmt->fetch();
52
53         if ($result['count'] > 0) {
54             return false; // Ya está matriculado
55         }
56
57         $sql = "INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date)
58             VALUES (:student_id, :course_id, :school_year_id, CURDATE())";
```

```

59
60     $stmt = $this->db->prepare($sql);
61     return $stmt->execute([
62         ':student_id' => $studentId,
63         ':course_id' => $courseId,
64         ':school_year_id' => $schoolYearId
65     ]);
66 }
67
68 public function getEnrolledStudents($courseId) {
69     $sql = "SELECT u.id, u.first_name, u.last_name, u.dni, cs.enrollment_date
70         FROM users u
71         INNER JOIN course_students cs ON u.id = cs.student_id
72         WHERE cs.course_id = :course_id
73         ORDER BY u.last_name, u.first_name";
74
75     $stmt = $this->db->prepare($sql);
76     $stmt->execute([':course_id' => $courseId]);
77     return $stmt->fetchAll();
78 }
79
80 public function findById($id) {
81     $sql = "SELECT c.*, s.name as shift_name
82         FROM courses c
83         INNER JOIN shifts s ON c.shift_id = s.id
84         WHERE c.id = :id";
85
86     $stmt = $this->db->prepare($sql);
87     $stmt->execute([':id' => $id]);
88     return $stmt->fetch();
89 }
90
91 public function update($data) {
92     $sql = "UPDATE courses SET
93         name = :name,
94         grade_level = :grade_level,
95         parallel = :parallel,
96         shift_id = :shift_id,
97         updated_at = NOW()
98         WHERE id = :id";
99
100     $stmt = $this->db->prepare($sql);
101     return $stmt->execute($data);
102 }
103
104 public function delete($id) {
105     // Eliminar relaciones primero
106     $sql = "DELETE FROM course_students WHERE course_id = :id";
107     $stmt = $this->db->prepare($sql);
108     $stmt->execute([':id' => $id]);
109
110     // Eliminar horarios
111     $sql = "DELETE FROM class_schedule WHERE course_id = :id";
112     $stmt = $this->db->prepare($sql);
113     $stmt->execute([':id' => $id]);
114
115     // Eliminar curso
116     $sql = "DELETE FROM courses WHERE id = :id";
117     $stmt = $this->db->prepare($sql);

```

```

118         return $stmt->execute([':id' => $id]);
119     }
120
121     public function unenrollStudent($studentId, $schoolYearId) {
122         $sql = "DELETE FROM course_students
123             WHERE student_id = :student_id AND school_year_id = :school_year_id";
124
125         $stmt = $this->db->prepare($sql);
126         return $stmt->execute([
127             ':student_id' => $studentId,
128             ':school_year_id' => $schoolYearId
129         ]);
130     }
131 }

```

38. ArchImp/models/Institution.php

```
1  <?php
2  class Institution {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getById($id) {
10         $sql = "SELECT * FROM institutions WHERE id = :id";
11         $stmt = $this->db->prepare($sql);
12         $stmt->execute([':id' => $id]);
13         return $stmt->fetch();
14     }
15
16     public function getAll() {
17         $sql = "SELECT * FROM institutions ORDER BY name";
18         $stmt = $this->db->query($sql);
19         return $stmt->fetchAll();
20     }
21
22     public function update($id, $data) {
23         $sql = "UPDATE institutions SET
24             name = :name,
25             address = :address,
26             province = :province,
27             city = :city,
28             phone = :phone,
29             email = :email,
30             director_name = :director_name,
31             amie_code = :amie_code,
32             website = :website,
33             logo_path = :logo_path,
34             updated_at = NOW()
35             WHERE id = :id";
36
37         // Obtener logo actual si no se proporciona uno nuevo
38         $currentInstitution = $this->getById($id);
39         $logoPath = $data['logo_path'] ?? $currentInstitution['logo_path'];
40
41         $stmt = $this->db->prepare($sql);
42         return $stmt->execute([
43             ':id' => $id,
44             ':name' => $data['name'],
45             ':address' => $data['address'],
46             ':province' => $data['province'],
47             ':city' => $data['city'],
48             ':phone' => $data['phone'],
49             ':email' => $data['email'],
50             ':director_name' => $data['director_name'],
51             ':amie_code' => $data['amie_code'],
52             ':website' => $data['website'],
53             ':logo_path' => $logoPath
54         ]);
55     }
56
57     public function getShifts($institutionId) {
58         $sql = "SELECT s.id, s.name,
```

```
59         GROUP_CONCAT(s.name ORDER BY s.name) as shift_names
60     FROM shifts s
61     INNER JOIN institution_shifts ins ON s.id = ins.shift_id
62     WHERE ins.institution_id = :institution_id
63     GROUP BY ins.institution_id";
64
65     $stmt = $this->db->prepare($sql);
66     $stmt->execute([':institution_id' => $institutionId]);
67     return $stmt->fetch();
68 }
69 }
```

39. ArchImp/models/InstitutionShift.php

```
1  <?php
2  class InstitutionShift {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function assign($data) {
10         $sql = "INSERT INTO institution_shifts (institution_id, shift_id)
11             VALUES (:institution_id, :shift_id)
12             ON DUPLICATE KEY UPDATE institution_id = institution_id";
13
14         $stmt = $this->db->prepare($sql);
15         return $stmt->execute($data);
16     }
17
18     public function remove($institutionId, $shiftId) {
19         $sql = "DELETE FROM institution_shifts
20             WHERE institution_id = :institution_id AND shift_id = :shift_id";
21
22         $stmt = $this->db->prepare($sql);
23         return $stmt->execute([
24             ':institution_id' => $institutionId,
25             ':shift_id' => $shiftId
26         ]);
27     }
28
29     public function getByInstitution($institutionId) {
30         $sql = "SELECT s.id, s.name
31             FROM shifts s
32             INNER JOIN institution_shifts ins ON s.id = ins.shift_id
33             WHERE ins.institution_id = :institution_id
34             ORDER BY s.name";
35
36         $stmt = $this->db->prepare($sql);
37         $stmt->execute([':institution_id' => $institutionId]);
38         return $stmt->fetchAll();
39     }
40
41     public function getInstitutionShiftIds($institutionId) {
42         $sql = "SELECT shift_id FROM institution_shifts WHERE institution_id = :institution_id";
43         $stmt = $this->db->prepare($sql);
44         $stmt->execute([':institution_id' => $institutionId]);
45         return $stmt->fetchAll(PDO::FETCH_COLUMN);
46     }
47 }
```


40. ArchImp/models/Justification.php

```
1  <?php
2  class Justification {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function create($data) {
10         $sql = "INSERT INTO justifications
11             (attendance_id, student_id, submitted_by, reason, document_path)
12             VALUES (:attendance_id, :student_id, :submitted_by, :reason, :document_path)";
13
14         $stmt = $this->db->prepare($sql);
15         return $stmt->execute($data);
16     }
17
18     public function getPending() {
19         $sql = "SELECT j.*,
20             CONCAT(u.last_name, ' ', u.first_name) as student_name,
21             CONCAT(s.last_name, ' ', s.first_name) as submitted_by_name,
22             a.date as attendance_date,
23             sub.name as subject_name,
24             c.name as course_name
25         FROM justifications j
26         INNER JOIN users u ON j.student_id = u.id
27         INNER JOIN users s ON j.submitted_by = s.id
28         INNER JOIN attendances a ON j.attendance_id = a.id
29         INNER JOIN subjects sub ON a.subject_id = sub.id
30         LEFT JOIN course_students cs ON u.id = cs.student_id
31         LEFT JOIN courses c ON cs.course_id = c.id
32         WHERE j.status = 'pendiente'
33         ORDER BY j.created_at DESC";
34
35         $stmt = $this->db->query($sql);
36         return $stmt->fetchAll();
37     }
38
39     public function getByStudent($studentId) {
40         $sql = "SELECT j.*,
41             a.date as attendance_date,
42             sub.name as subject_name,
43             j.created_at as submitted_at,
44             j.updated_at as reviewed_at,
45             CONCAT(r.last_name, ' ', r.first_name) as reviewer_name
46         FROM justifications j
47         INNER JOIN attendances a ON j.attendance_id = a.id
48         INNER JOIN subjects sub ON a.subject_id = sub.id
49         LEFT JOIN users r ON j.reviewed_by = r.id
50         WHERE j.student_id = :student_id
51         ORDER BY j.created_at DESC";
52
53         $stmt = $this->db->prepare($sql);
54         $stmt->execute([':student_id' => $studentId]);
55         return $stmt->fetchAll();
56     }
57
58     public function approve($justificationId, $reviewerId, $notes = '') {
```

```

59     $sql = "UPDATE justifications
60         SET status = 'aprobado',
61             reviewed_by = :reviewed_by,
62             review_notes = :notes,
63             updated_at = NOW()
64         WHERE id = :id";
65
66     $stmt = $this->db->prepare($sql);
67     $result = $stmt->execute([
68         ':reviewed_by' => $reviewerId,
69         ':notes' => $notes,
70         ':id' => $justificationId
71     ]);
72
73     if ($result) {
74         // Actualizar asistencia a justificado
75         $sql2 = "UPDATE attendances a
76             INNER JOIN justifications j ON a.id = j.attendance_id
77             SET a.status = 'justificado'
78             WHERE j.id = :id";
79         $stmt2 = $this->db->prepare($sql2);
80         $stmt2->execute([':id' => $justificationId]);
81     }
82
83     return $result;
84 }
85
86 public function reject($justificationId, $reviewerId, $notes) {
87     $sql = "UPDATE justifications
88         SET status = 'rechazado',
89             reviewed_by = :reviewed_by,
90             review_notes = :notes,
91             updated_at = NOW()
92         WHERE id = :id";
93
94     $stmt = $this->db->prepare($sql);
95     return $stmt->execute([
96         ':reviewed_by' => $reviewerId,
97         ':notes' => $notes,
98         ':id' => $justificationId
99     ]);
100 }
101
102 public function existsByAttendance($attendanceId) {
103     $sql = "SELECT COUNT(*) as count FROM justifications WHERE attendance_id = :attendance_id";
104     $stmt = $this->db->prepare($sql);
105     $stmt->execute([':attendance_id' => $attendanceId]);
106     $result = $stmt->fetch();
107     return $result['count'] > 0;
108 }
109
110 public function getById($id) {
111     $sql = "SELECT j.*, j.student_id, j.submitted_by,
112         CONCAT(u.last_name, ' ', u.first_name) as student_name
113     FROM justifications j
114     INNER JOIN users u ON j.student_id = u.id
115     WHERE j.id = :id";
116     $stmt = $this->db->prepare($sql);
117     $stmt->execute([':id' => $id]);

```

[illegible]

[illegible]

```

236     });
237
238     if ($result) {
239         // Si tiene attendance_id único, actualizar también esa asistencia
240         $j = $this->getById($justificationId);
241         if (!empty($j['attendance_id'])) {
242             $this->db->prepare(
243                 "UPDATE attendances SET status='justificado' WHERE id=:id"
244             )->execute([':id' => $j['attendance_id']]);
245         }
246         // Si tiene rango de fechas, actualizar todas las ausencias del estudiante en ese rango
247         if (!empty($j['date_from']) && !empty($j['date_to'])) {
248             $this->db->prepare(
249                 "UPDATE attendances SET status='justificado'
250                 WHERE student_id=:sid AND date BETWEEN :df AND :dt AND status='ausente'"
251             )->execute([
252                 ':sid' => $j['student_id'],
253                 ':df' => $j['date_from'],
254                 ':dt' => $j['date_to'],
255             ]);
256         }
257     }
258     return $result;
259 }
260
261 // Crear una justificación por cada asistencia seleccionada
262 public function createForAttendances($attendanceIds, $data) {
263     $sql = "INSERT INTO justifications
264         (attendance_id, student_id, submitted_by,
265         date_from, date_to, working_days,
266         reason_type, reason, document_path, can_approve, status)
267     VALUES
268         (:attendance_id, :student_id, :submitted_by,
269         :date_from, :date_to, :working_days,
270         :reason_type, :reason, :document_path, :can_approve, 'pendiente)";
271     $stmt = $this->db->prepare($sql);
272     $ok = true;
273     foreach ($attendanceIds as $attId) {
274         $row = $data;
275         $row[':attendance_id'] = (int)$attId;
276         if (!$stmt->execute($row)) $ok = false;
277     }
278     return $ok;
279 }
280
281 // Contar ausencias ya justificadas (pendiente o aprobada) para un estudiante en un rango
282 public function countPendingJustifications($studentId) {
283     $sql = "SELECT COUNT(*) FROM justifications
284         WHERE student_id = :sid AND status = 'pendiente'";
285     $stmt = $this->db->prepare($sql);
286     $stmt->execute([':sid' => $studentId]);
287     return (int)$stmt->fetchColumn();
288 }
289 }

```

41. ArchImp/models/Notification.php

```
1  <?php
2  class Notification {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      // Crear notificación
10     // $link: URL relativa destino, ej: '?action=pending_justifications'
11     public function create($userId, $title, $message, $type = 'info', $link = null) {
12         $sql = "INSERT INTO notifications (user_id, title, message, type, link, is_read, created_at)
13             VALUES (:user_id, :title, :message, :type, :link, 0, NOW())";
14         $stmt = $this->db->prepare($sql);
15         return $stmt->execute([
16             ':user_id' => $userId,
17             ':title'   => $title,
18             ':message' => $message,
19             ':type'    => $type,
20             ':link'    => $link
21         ]);
22     }
23
24     // Obtener notificaciones de un usuario (con paginación)
25     public function getByUser($userId, $limit = 20, $offset = 0) {
26         $sql = "SELECT * FROM notifications
27             WHERE user_id = :user_id
28             ORDER BY created_at DESC
29             LIMIT :limit OFFSET :offset";
30         $stmt = $this->db->prepare($sql);
31         $stmt->bindValue(':user_id', $userId, PDO::PARAM_INT);
32         $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
33         $stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
34         $stmt->execute();
35         return $stmt->fetchAll();
36     }
37
38     // Solo no leídas (para el dropdown del navbar)
39     public function getUnread($userId, $limit = 10) {
40         $sql = "SELECT * FROM notifications
41             WHERE user_id = :user_id AND is_read = 0
42             ORDER BY created_at DESC
43             LIMIT :limit";
44         $stmt = $this->db->prepare($sql);
45         $stmt->bindValue(':user_id', $userId, PDO::PARAM_INT);
46         $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
47         $stmt->execute();
48         return $stmt->fetchAll();
49     }
50
51     // Contar no leídas
52     public function getUnreadCount($userId) {
53         $sql = "SELECT COUNT(*) as count FROM notifications
54             WHERE user_id = :user_id AND is_read = 0";
55         $stmt = $this->db->prepare($sql);
56         $stmt->execute([':user_id' => $userId]);
57         return (int)$stmt->fetch()['count'];
58     }
59 }
```

```

59
60 // Marcar una como leída
61 public function markAsRead($notificationId, $userId) {
62     $sql = "UPDATE notifications SET is_read = 1
63         WHERE id = :id AND user_id = :user_id";
64     $stmt = $this->db->prepare($sql);
65     return $stmt->execute([':id' => $notificationId, ':user_id' => $userId]);
66 }
67
68 // Marcar TODAS como leídas
69 public function markAllAsRead($userId) {
70     $sql = "UPDATE notifications SET is_read = 1
71         WHERE user_id = :user_id AND is_read = 0";
72     $stmt = $this->db->prepare($sql);
73     return $stmt->execute([':user_id' => $userId]);
74 }
75
76 // Eliminar una notificación
77 public function delete($notificationId, $userId) {
78     $sql = "DELETE FROM notifications WHERE id = :id AND user_id = :user_id";
79     $stmt = $this->db->prepare($sql);
80     return $stmt->execute([':id' => $notificationId, ':user_id' => $userId]);
81 }
82
83 // Eliminar todas las leídas
84 public function deleteRead($userId) {
85     $sql = "DELETE FROM notifications WHERE user_id = :user_id AND is_read = 1";
86     $stmt = $this->db->prepare($sql);
87     return $stmt->execute([':user_id' => $userId]);
88 }
89
90 // Total de notificaciones del usuario (para paginación)
91 public function countByUser($userId) {
92     $sql = "SELECT COUNT(*) as count FROM notifications WHERE user_id = :user_id";
93     $stmt = $this->db->prepare($sql);
94     $stmt->execute([':user_id' => $userId]);
95     return (int)$stmt->fetch()['count'];
96 }
97 }

```

42. ArchImp/models/Representative.php

```
1  <?php
2  class Representative {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function assignStudent($representativeId, $studentId, $relationship, $isPrimary = 0) {
10         $sql = "INSERT INTO representatives
11             (representative_id, student_id, relationship, is_primary)
12             VALUES
13             (:rep_id, :stu_id, :rel_ins, :prim_ins)
14             ON DUPLICATE KEY UPDATE
15             relationship = :rel_upd,
16             is_primary = :prim_upd";
17
18         $stmt = $this->db->prepare($sql);
19
20         return $stmt->execute([
21             ':rep_id'   => $representativeId,
22             ':stu_id'   => $studentId,
23             ':rel_ins'  => $relationship,
24             ':prim_ins' => $isPrimary,
25             ':rel_upd'  => $relationship,
26             ':prim_upd' => $isPrimary
27         ]);
28     }
29
30     public function getStudentsByRepresentative($representativeId) {
31         $sql = "SELECT u.*, r.relationship, r.is_primary, c.name as course_name, sh.name as shift_name
32             FROM users u
33             INNER JOIN representatives r ON u.id = r.student_id
34             LEFT JOIN course_students cs ON u.id = cs.student_id
35             LEFT JOIN courses c ON cs.course_id = c.id
36             LEFT JOIN shifts sh ON c.shift_id = sh.id
37             WHERE r.representative_id = :representative_id
38             ORDER BY r.is_primary DESC, u.last_name, u.first_name";
39
40         $stmt = $this->db->prepare($sql);
41         $stmt->execute([':representative_id' => $representativeId]);
42         return $stmt->fetchAll();
43     }
44
45     public function getRepresentativesByStudent($studentId) {
46         $sql = "SELECT u.*, r.relationship, r.is_primary
47             FROM users u
48             INNER JOIN representatives r ON u.id = r.representative_id
49             WHERE r.student_id = :student_id
50             ORDER BY r.is_primary DESC";
51
52         $stmt = $this->db->prepare($sql);
53         $stmt->execute([':student_id' => $studentId]);
54         return $stmt->fetchAll();
55     }
56
57     public function removeStudent($representativeId, $studentId) {
58         $sql = "DELETE FROM representatives
```



```
59             WHERE representative_id = :representative_id AND student_id = :student_id";
60
61     $stmt = $this->db->prepare($sql);
62     return $stmt->execute([
63         ':representative_id' => $representativeId,
64         ':student_id' => $studentId
65     ]);
66 }
67 }
```

43. ArchImp/models/Role.php

```
1  <?php
2  class Role {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getAll() {
10         $sql = "SELECT * FROM roles ORDER BY name";
11         $stmt = $this->db->query($sql);
12         return $stmt->fetchAll();
13     }
14
15     public function findById($id) {
16         $sql = "SELECT * FROM roles WHERE id = :id";
17         $stmt = $this->db->prepare($sql);
18         $stmt->execute([':id' => $id]);
19         return $stmt->fetch();
20     }
21 }
```

44. ArchImp/models/SchoolYear.php

```
1  <?php
2  class SchoolYear {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getActive() {
10         $sql = "SELECT * FROM school_years
11             WHERE institution_id = :institution_id AND is_active = 1
12             LIMIT 1";
13
14         $stmt = $this->db->prepare($sql);
15         $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
16         return $stmt->fetch();
17     }
18
19     public function getAll() {
20         $sql = "SELECT * FROM school_years
21             WHERE institution_id = :institution_id
22             ORDER BY start_date DESC";
23
24         $stmt = $this->db->prepare($sql);
25         $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
26         return $stmt->fetchAll();
27     }
28
29
30     public function findById($id) {
31         $sql = "SELECT * FROM school_years
32             WHERE id = :id AND institution_id = :institution_id";
33
34         $stmt = $this->db->prepare($sql);
35         $stmt->execute([
36             ':id' => $id,
37             ':institution_id' => $_SESSION['institution_id']
38         ]);
39
40         return $stmt->fetch();
41     }
42
43     public function create($data) {
44         $sql = "INSERT INTO school_years
45             (institution_id, name, start_date, end_date, is_active)
46             VALUES
47             (:institution_id, :name, :start_date, :end_date, :is_active)";
48
49         $stmt = $this->db->prepare($sql);
50         $ok = $stmt->execute([
51             ':institution_id' => $data['institution_id'],
52             ':name'           => $data['name'],
53             ':start_date'     => $data['start_date'],
54             ':end_date'       => $data['end_date'],
55             ':is_active'      => $data['is_active']
56         ]);
57         // Retorna el ID insertado para poder activarlo en la misma conexión
58         return $ok ? (int)$this->db->lastInsertId() : 0;
```

```

59     }
60
61     public function update($data) {
62         $sql = "UPDATE school_years
63             SET name=:name,
64                 start_date=:start_date,
65                 end_date=:end_date
66             WHERE id=:id
67                 AND institution_id=:institution_id";
68
69         $stmt = $this->db->prepare($sql);
70
71         return $stmt->execute([
72             ':id'=>$data['id'],
73             ':name'=>$data['name'],
74             ':start_date'=>$data['start_date'],
75             ':end_date'=>$data['end_date'],
76             ':institution_id'=>$SESSION['institution_id']
77         ]);
78     }
79
80     public function delete($id) {
81         $stmt = $this->db->prepare(
82             "DELETE FROM school_years
83             WHERE id=:id AND institution_id=:institution_id"
84         );
85
86         return $stmt->execute([
87             ':id'=>$id,
88             ':institution_id'=>$SESSION['institution_id']
89         ]);
90     }
91
92     public function activate($id) {
93
94         // desactiva todos
95         $this->db->prepare(
96             "UPDATE school_years
97             SET is_active=0
98             WHERE institution_id=:institution_id"
99         )->execute([':institution_id'=>$SESSION['institution_id']]);
100
101         if ($id == 0) return true;
102
103         // activa uno
104         $stmt = $this->db->prepare(
105             "UPDATE school_years
106             SET is_active=1
107             WHERE id=:id AND institution_id=:institution_id"
108         );
109
110         return $stmt->execute([
111             ':id'=>$id,
112             ':institution_id'=>$SESSION['institution_id']
113         ]);
114     }
115
116     public function deactivate($id) {
117         $stmt = $this->db->prepare(

```

```

118         "UPDATE school_years
119         SET is_active=0
120         WHERE id=:id AND institution_id=:institution_id"
121     );
122
123     return $stmt->execute([
124         ':id'=>$id,
125         ':institution_id'=>$SESSION['institution_id']
126     ]);
127 }
128
129 public function checkOverlap($start, $end, $excludeId = null) {
130
131     $sql = "SELECT COUNT(*)
132     FROM school_years
133     WHERE institution_id = :institution_id
134     AND (start_date <= :end AND end_date >= :start)";
135
136     if ($excludeId) {
137         $sql .= " AND id != :excludeId";
138     }
139
140     $stmt = $this->db->prepare($sql);
141
142     $params = [
143         ':institution_id' => $SESSION['institution_id'],
144         ':start' => $start,
145         ':end' => $end
146     ];
147
148     if ($excludeId) {
149         $params[':excludeId'] = $excludeId;
150     }
151
152     $stmt->execute($params);
153     return $stmt->fetchColumn() > 0;
154 }
155
156 }

```

45. ArchImp/models/Shift.php

```
1  <?php
2  class Shift {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getAll() {
10         $sql = "SELECT * FROM shifts ORDER BY id";
11         $stmt = $this->db->query($sql);
12         return $stmt->fetchAll();
13     }
14 }
```

46. ArchImp/models/Subject.php

```
1  <?php
2  class Subject {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function getAll() {
10         $sql = "SELECT * FROM subjects
11             WHERE institution_id = :institution_id
12             ORDER BY name";
13
14         $stmt = $this->db->prepare($sql);
15         $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
16         return $stmt->fetchAll();
17     }
18
19     public function create($data) {
20         $sql = "INSERT INTO subjects (institution_id, name, code)
21             VALUES (:institution_id, :name, :code)";
22
23         $stmt = $this->db->prepare($sql);
24         return $stmt->execute($data);
25     }
26
27     public function findById($id) {
28         $sql = "SELECT * FROM subjects WHERE id = :id";
29         $stmt = $this->db->prepare($sql);
30         $stmt->execute([':id' => $id]);
31         return $stmt->fetch();
32     }
33
34     public function update($data) {
35         $sql = "UPDATE subjects SET
36             name = :name,
37             code = :code,
38             updated_at = NOW()
39             WHERE id = :id";
40
41         $stmt = $this->db->prepare($sql);
42         return $stmt->execute($data);
43     }
44
45     public function delete($id) {
46         $sql = "DELETE FROM subjects WHERE id = :id";
47         $stmt = $this->db->prepare($sql);
48         return $stmt->execute([':id' => $id]);
49     }
50 }
```

47. ArchImp/models/TeacherAssignment.php

```
1  <?php
2  class TeacherAssignment {
3      private $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function assign($data) {
10         // Verificar si ya existe asignación curso-materia
11         $checkSql = "SELECT ta.id, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
12             FROM teacher_assignments ta
13             INNER JOIN users u ON ta.teacher_id = u.id
14             WHERE ta.course_id = :course_id
15             AND ta.subject_id = :subject_id
16             AND ta.school_year_id = :school_year_id";
17
18         $checkStmt = $this->db->prepare($checkSql);
19         $checkStmt->execute([
20             ':course_id' => $data[':course_id'],
21             ':subject_id' => $data[':subject_id'],
22             ':school_year_id' => $data[':school_year_id']
23         ]);
24
25         $existing = $checkStmt->fetch();
26
27         if ($existing) {
28             return [
29                 'success' => false,
30                 'message' => 'La materia ya está asignada a: ' . $existing['teacher_name']
31             ];
32         }
33
34         $sql = "INSERT INTO teacher_assignments
35             (teacher_id, course_id, subject_id, school_year_id, is_tutor)
36             VALUES (:teacher_id, :course_id, :subject_id, :school_year_id, :is_tutor)";
37
38         $stmt = $this->db->prepare($sql);
39         $result = $stmt->execute($data);
40
41         return [
42             'success' => $result,
43             'message' => 'Asignación creada correctamente'
44         ];
45     }
46
47     public function getByTeacher($teacherId) {
48         $sql = "SELECT ta.*, c.name as course_name, s.name as subject_name,
49             sh.name as shift_name, sy.name as year_name
50             FROM teacher_assignments ta
51             INNER JOIN courses c ON ta.course_id = c.id
52             INNER JOIN subjects s ON ta.subject_id = s.id
53             INNER JOIN shifts sh ON c.shift_id = sh.id
54             INNER JOIN school_years sy ON ta.school_year_id = sy.id
55             WHERE ta.teacher_id = :teacher_id
56             ORDER BY c.name, s.name";
57
58         $stmt = $this->db->prepare($sql);
```



```

59     $stmt->execute([':teacher_id' => $teacherId]);
60     return $stmt->fetchAll();
61 }
62
63 public function getByCourse($courseId) {
64     $sql = "SELECT ta.*,
65             CONCAT(u.last_name, ' ', u.first_name) as teacher_name,
66             s.name as subject_name
67             FROM teacher_assignments ta
68             INNER JOIN users u ON ta.teacher_id = u.id
69             INNER JOIN subjects s ON ta.subject_id = s.id
70             WHERE ta.course_id = :course_id
71             ORDER BY s.name";
72
73     $stmt = $this->db->prepare($sql);
74     $stmt->execute([':course_id' => $courseId]);
75     return $stmt->fetchAll();
76 }
77
78 public function getTutorByCourse($courseId, $schoolYearId) {
79     $sql = "SELECT u.*, ta.id as assignment_id
80             FROM teacher_assignments ta
81             INNER JOIN users u ON ta.teacher_id = u.id
82             WHERE ta.course_id = :course_id
83             AND ta.school_year_id = :school_year_id
84             AND ta.is_tutor = 1
85             LIMIT 1";
86
87     $stmt = $this->db->prepare($sql);
88     $stmt->execute([
89         ':course_id' => $courseId,
90         ':school_year_id' => $schoolYearId
91     ]);
92     return $stmt->fetch();
93 }
94
95 public function setTutor($courseId, $teacherId, $schoolYearId) {
96     // Verificar si el docente ya es tutor de otro curso en el mismo año lectivo
97     $checkSql = "SELECT c.name
98                 FROM teacher_assignments ta
99                 INNER JOIN courses c ON ta.course_id = c.id
100                WHERE ta.teacher_id = :teacher_id
101                AND ta.school_year_id = :school_year_id
102                AND ta.is_tutor = 1
103                AND ta.course_id != :course_id";
104
105     $checkStmt = $this->db->prepare($checkSql);
106     $checkStmt->execute([
107         ':teacher_id' => $teacherId,
108         ':school_year_id' => $schoolYearId,
109         ':course_id' => $courseId
110     ]);
111
112     $existingTutor = $checkStmt->fetch();
113
114     if ($existingTutor) {
115         return [
116             'success' => false,
117             'message' => 'Este docente ya es tutor del curso: ' . $existingTutor['name']

```

```

118         ];
119     }
120
121     // Quitar tutor actual del curso
122     $sql1 = "UPDATE teacher_assignments
123         SET is_tutor = 0
124         WHERE course_id = :course_id AND school_year_id = :school_year_id";
125     $stmt1 = $this->db->prepare($sql1);
126     $stmt1->execute([
127         ':course_id' => $courseId,
128         ':school_year_id' => $schoolYearId
129     ]);
130
131     // Verificar si docente tiene asignación en el curso
132     $checkAssignment = "SELECT id FROM teacher_assignments
133         WHERE course_id = :course_id
134         AND teacher_id = :teacher_id
135         AND school_year_id = :school_year_id
136         LIMIT 1";
137     $stmt = $this->db->prepare($checkAssignment);
138     $stmt->execute([
139         ':course_id' => $courseId,
140         ':teacher_id' => $teacherId,
141         ':school_year_id' => $schoolYearId
142     ]);
143     $existing = $stmt->fetch();
144
145     if ($existing) {
146         // Actualizar asignación existente
147         $sql2 = "UPDATE teacher_assignments
148             SET is_tutor = 1
149             WHERE id = :id";
150         $stmt2 = $this->db->prepare($sql2);
151         $result = $stmt2->execute([':id' => $existing['id']]);
152     } else {
153         return [
154             'success' => false,
155             'message' => 'El docente debe tener al menos una asignatura en el curso antes de ser tutor'
156         ];
157     }
158
159     return [
160         'success' => $result,
161         'message' => 'Tutor asignado correctamente'
162     ];
163 }
164
165 public function remove($assignmentId) {
166     // Verificar si es tutor antes de eliminar
167     $checkSql = "SELECT is_tutor FROM teacher_assignments WHERE id = :id";
168     $checkStmt = $this->db->prepare($checkSql);
169     $checkStmt->execute([':id' => $assignmentId]);
170     $assignment = $checkStmt->fetch();
171
172     if ($assignment && $assignment['is_tutor'] == 1) {
173         return [
174             'success' => false,
175             'message' => 'No se puede eliminar: este docente es tutor del curso. Quite primero la tutoría.'
176         ];

```

```

177     }
178
179     $sql = "DELETE FROM teacher_assignments WHERE id = :id";
180     $stmt = $this->db->prepare($sql);
181     $result = $stmt->execute([':id' => $assignmentId]);
182
183     return [
184         'success' => $result,
185         'message' => 'Asignación eliminada correctamente'
186     ];
187 }
188
189 public function getAll() {
190     $sql = "SELECT ta.*,
191         CONCAT(u.last_name, ' ', u.first_name) as teacher_name,
192         c.name as course_name,
193         s.name as subject_name,
194         sy.name as year_name
195     FROM teacher_assignments ta
196     INNER JOIN users u ON ta.teacher_id = u.id
197     INNER JOIN courses c ON ta.course_id = c.id
198     INNER JOIN subjects s ON ta.subject_id = s.id
199     INNER JOIN school_years sy ON ta.school_year_id = sy.id
200     ORDER BY c.name, s.name";
201
202     $stmt = $this->db->query($sql);
203     return $stmt->fetchAll();
204 }
205 }

```

48. ArchImp/models/User.php

```
1  <?php
2  class User {
3      public $db;
4
5      public function __construct($database) {
6          $this->db = $database->connect();
7      }
8
9      public function create($data) {
10         $sql = "INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone)
11             VALUES (:institution_id, :username, :email, :password, :first_name, :last_name, :dni, :phone)";
12
13         $stmt = $this->db->prepare($sql);
14         return $stmt->execute([
15             ':institution_id' => $data['institution_id'],
16             ':username' => $data['username'],
17             ':email' => $data['email'],
18             ':password' => Security::hashPassword($data['password']),
19             ':first_name' => $data['first_name'],
20             ':last_name' => $data['last_name'],
21             ':dni' => $data['dni'] ?? null,
22             ':phone' => $data['phone'] ?? null
23         ]);
24     }
25
26     public function findByEmail($email) {
27         $sql = "SELECT * FROM users WHERE email = :email AND is_active = 1";
28         $stmt = $this->db->prepare($sql);
29         $stmt->execute([':email' => $email]);
30         return $stmt->fetch();
31     }
32
33     public function findById($id) {
34         $sql = "SELECT * FROM users WHERE id = :id";
35         $stmt = $this->db->prepare($sql);
36         $stmt->execute([':id' => $id]);
37         return $stmt->fetch();
38     }
39
40     public function getUserRoles($userId) {
41         $sql = "SELECT r.name FROM roles r
42             INNER JOIN user_roles ur ON r.id = ur.role_id
43             WHERE ur.user_id = :user_id";
44         $stmt = $this->db->prepare($sql);
45         $stmt->execute([':user_id' => $userId]);
46         return $stmt->fetchAll(PDO::FETCH_COLUMN);
47     }
48
49     public function assignRole($userId, $roleId) {
50         $sql = "INSERT IGNORE INTO user_roles (user_id, role_id) VALUES (:user_id, :role_id)";
51         $stmt = $this->db->prepare($sql);
52         return $stmt->execute([':user_id' => $userId, ':role_id' => $roleId]);
53     }
54
55     public function setResetToken($email, $token) {
56         $sql = "UPDATE users SET reset_token = :token, reset_expires = DATE_ADD(NOW(), INTERVAL 1 HOUR)
57             WHERE email = :email";
58         $stmt = $this->db->prepare($sql);
```

```

59         return $stmt->execute([':token' => $token, ':email' => $email]);
60     }
61
62     public function validateResetToken($token) {
63         $sql = "SELECT * FROM users WHERE reset_token = :token AND reset_expires > NOW()";
64         $stmt = $this->db->prepare($sql);
65         $stmt->execute([':token' => $token]);
66         return $stmt->fetch();
67     }
68
69     public function resetPassword($userId, $newPassword) {
70         $sql = "UPDATE users SET password = :password, reset_token = NULL, reset_expires = NULL
71             WHERE id = :id";
72         $stmt = $this->db->prepare($sql);
73         return $stmt->execute([
74             ':password' => Security::hashPassword($newPassword),
75             ':id' => $userId
76         ]);
77     }
78
79     public function getAll() {
80         $sql = "SELECT u.*, GROUP_CONCAT(r.name) as roles
81             FROM users u
82             LEFT JOIN user_roles ur ON u.id = ur.user_id
83             LEFT JOIN roles r ON ur.role_id = r.id
84             WHERE u.institution_id = :institution_id
85             GROUP BY u.id
86             ORDER BY u.last_name, u.first_name";
87
88         $stmt = $this->db->prepare($sql);
89         $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
90         return $stmt->fetchAll();
91     }
92
93     public function removeRole($userId, $roleId) {
94         $sql = "DELETE FROM user_roles WHERE user_id = :user_id AND role_id = :role_id";
95         $stmt = $this->db->prepare($sql);
96         return $stmt->execute([':user_id' => $userId, ':role_id' => $roleId]);
97     }
98
99     public function getUserWithRoles($userId) {
100         $user = $this->findById($userId);
101         if ($user) {
102             $user['roles'] = $this->getUserRoles($userId);
103         }
104         return $user;
105     }
106
107     /* public function getByRole($roleName) {
108         $sql = "SELECT u.* FROM users u
109             INNER JOIN user_roles ur ON u.id = ur.user_id
110             INNER JOIN roles r ON ur.role_id = r.id
111             WHERE r.name = :role_name AND u.institution_id = :institution_id
112             ORDER BY u.last_name, u.first_name";
113
114         $stmt = $this->db->prepare($sql);
115         $stmt->execute([
116             ':role_name' => $roleName,
117             ':institution_id' => $_SESSION['institution_id']

```

```

118         });
119         return $stmt->fetchAll();
120     } */
121
122     // Agregar al final de la clase User
123
124     public function getByRole($roleName) {
125         $sql = "SELECT u.* FROM users u
126             INNER JOIN user_roles ur ON u.id = ur.user_id
127             INNER JOIN roles r ON ur.role_id = r.id
128             WHERE r.name = :role_name AND u.institution_id = :institution_id
129             ORDER BY u.last_name, u.first_name";
130
131         $stmt = $this->db->prepare($sql);
132         $stmt->execute([
133             ':role_name' => $roleName,
134             ':institution_id' => $_SESSION['institution_id']
135         ]);
136         return $stmt->fetchAll();
137     }
138
139     public function getStudentsNotEnrolled($schoolYearId) {
140         $sql = "SELECT u.* FROM users u
141             INNER JOIN user_roles ur ON u.id = ur.user_id
142             INNER JOIN roles r ON ur.role_id = r.id
143             WHERE r.name = 'estudiante'
144             AND u.institution_id = :institution_id
145             AND u.id NOT IN (
146                 SELECT student_id FROM course_students
147                 WHERE school_year_id = :school_year_id
148             )
149             ORDER BY u.last_name, u.first_name";
150
151         $stmt = $this->db->prepare($sql);
152         $stmt->execute([
153             ':institution_id' => $_SESSION['institution_id'],
154             ':school_year_id' => $schoolYearId
155         ]);
156         return $stmt->fetchAll();
157     }
158
159     public function getStudentCourse($studentId, $schoolYearId) {
160         $sql = "SELECT c.*, s.name as shift_name
161             FROM courses c
162             INNER JOIN course_students cs ON c.id = cs.course_id
163             INNER JOIN shifts s ON c.shift_id = s.id
164             WHERE cs.student_id = :student_id
165             AND cs.school_year_id = :school_year_id";
166
167         $stmt = $this->db->prepare($sql);
168         $stmt->execute([
169             ':student_id' => $studentId,
170             ':school_year_id' => $schoolYearId
171         ]);
172         return $stmt->fetch();
173     }
174
175     public function updateProfile($userId, $data) {
176         $sql = "UPDATE users

```

```

177         SET first_name = :first_name,
178             last_name = :last_name,
179             phone = :phone,
180             email = :email,
181             updated_at = NOW()
182     WHERE id = :id";
183
184     $stmt = $this->db->prepare($sql);
185     return $stmt->execute([
186         ':first_name' => $data['first_name'],
187         ':last_name' => $data['last_name'],
188         ':phone' => $data['phone'],
189         ':email' => $data['email'],
190         ':id' => $userId
191     ]);
192 }
193
194 public function updatePassword($userId, $newPassword) {
195     $sql = "UPDATE users SET password = :password WHERE id = :id";
196     $stmt = $this->db->prepare($sql);
197     return $stmt->execute([
198         ':password' => Security::hashPassword($newPassword),
199         ':id' => $userId
200     ]);
201 }
202
203 public function findByEmailOrUsername($emailOrUsername) {
204     $sql = "SELECT * FROM users
205         WHERE (email = :email OR username = :username)
206         AND is_active = 1";
207
208     $stmt = $this->db->prepare($sql);
209     $stmt->execute([
210         ':email' => $emailOrUsername,
211         ':username' => $emailOrUsername
212     ]);
213     return $stmt->fetch();
214 }
215
216 public function getUserRoleIds($userId) {
217     $sql = "SELECT r.id FROM roles r
218         INNER JOIN user_roles ur ON r.id = ur.role_id
219         WHERE ur.user_id = :user_id
220         ORDER BY r.name";
221     $stmt = $this->db->prepare($sql);
222     $stmt->execute([':user_id' => $userId]);
223     return $stmt->fetchAll(PDO::FETCH_COLUMN);
224 }
225
226 public function findByUsername($username) {
227     $sql = "SELECT * FROM users WHERE username = :username";
228     $stmt = $this->db->prepare($sql);
229     $stmt->execute([':username' => $username]);
230     return $stmt->fetch();
231 }
232
233 public function update($data) {
234     // Si se incluye password, actualizarla
235     if (isset($data['password'])) {

```

```

236     $sql = "UPDATE users SET
237         username = :username,
238         email = :email,
239         password = :password,
240         first_name = :first_name,
241         last_name = :last_name,
242         dni = :dni,
243         phone = :phone,
244         updated_at = NOW()
245         WHERE id = :id";
246
247     $params = [
248         ':id' => $data['id'],
249         ':username' => $data['username'],
250         ':email' => $data['email'],
251         ':password' => Security::hashPassword($data['password']),
252         ':first_name' => $data['first_name'],
253         ':last_name' => $data['last_name'],
254         ':dni' => $data['dni'],
255         ':phone' => $data['phone']
256     ];
257 } else {
258     // No actualizar password
259     $sql = "UPDATE users SET
260         username = :username,
261         email = :email,
262         first_name = :first_name,
263         last_name = :last_name,
264         dni = :dni,
265         phone = :phone,
266         updated_at = NOW()
267         WHERE id = :id";
268
269     $params = [
270         ':id' => $data['id'],
271         ':username' => $data['username'],
272         ':email' => $data['email'],
273         ':first_name' => $data['first_name'],
274         ':last_name' => $data['last_name'],
275         ':dni' => $data['dni'],
276         ':phone' => $data['phone']
277     ];
278 }
279
280 $stmt = $this->db->prepare($sql);
281 return $stmt->execute($params);
282 }
283
284 public function delete($userId) {
285     // Primero eliminar roles
286     $sql = "DELETE FROM user_roles WHERE user_id = :user_id";
287     $stmt = $this->db->prepare($sql);
288     $stmt->execute([':user_id' => $userId]);
289
290     // Eliminar representaciones (tanto como representante como estudiante)
291     $sql = "DELETE FROM representatives
292         WHERE representative_id = :rep_id OR student_id = :stu_id";
293     $stmt = $this->db->prepare($sql);
294     $stmt->execute([

```



```

295         ':rep_id' => $userId,
296         ':stu_id' => $userId
297     });
298
299     // Eliminar de course_students
300     $sql = "DELETE FROM course_students WHERE student_id = :user_id";
301     $stmt = $this->db->prepare($sql);
302     $stmt->execute([':user_id' => $userId]);
303
304     // Eliminar asignaciones docentes
305     $sql = "DELETE FROM teacher_assignments WHERE teacher_id = :user_id";
306     $stmt = $this->db->prepare($sql);
307     $stmt->execute([':user_id' => $userId]);
308
309     // Finalmente eliminar usuario
310     $sql = "DELETE FROM users WHERE id = :id";
311     $stmt = $this->db->prepare($sql);
312     return $stmt->execute([':id' => $userId]);
313 }
314
315 public function deactivate($userId) {
316     $sql = "UPDATE users SET is_active = 0, updated_at = NOW() WHERE id = :id";
317     $stmt = $this->db->prepare($sql);
318     return $stmt->execute([':id' => $userId]);
319 }
320 }

```

49. ArchImp/public/.htaccess

```
1 <IfModule mod_rewrite.c>
2     RewriteEngine On
3     RewriteBase /ecuasistencia2026/public/
4
5     RewriteCond %{REQUEST_FILENAME} !-f
6     RewriteCond %{REQUEST_FILENAME} !-d
7     RewriteRule ^(.*)$ index.php?action=$1 [QSA,L]
8 </IfModule>
```

50. ArchImp/public/img.php

```
1  <?php
2  // public/img.php - Sirve imágenes desde uploads/ (fuera del document root)
3  // Depende de env.php para obtener BASE_PATH correcto en local y producción
4
5  require_once __DIR__ . '/../config/env.php';
6
7  $uploadsDir = realpath(BASE_PATH . '/uploads');
8
9  if (!$uploadsDir) {
10     http_response_code(404); exit;
11 }
12
13 $file = $_GET['f'] ?? '';
14
15 // Seguridad: bloquear path traversal
16 $file = str_replace(['..', '\\', "\0"], '', $file);
17 $file = ltrim($file, '/');
18
19 if (!$file) { http_response_code(400); exit; }
20
21 $fullPath = $uploadsDir . DIRECTORY_SEPARATOR . $file;
22 $realFull = realpath($fullPath);
23
24 // Verificar que el archivo está dentro de uploads/
25 if (!$realFull || strpos($realFull, $uploadsDir) !== 0) {
26     http_response_code(403); exit;
27 }
28
29 if (!file_exists($realFull) || !is_file($realFull)) {
30     http_response_code(404); exit;
31 }
32
33 $ext = strtolower(pathinfo($realFull, PATHINFO_EXTENSION));
34 $mimes = [
35     'jpg' => 'image/jpeg',
36     'jpeg' => 'image/jpeg',
37     'png' => 'image/png',
38     'gif' => 'image/gif',
39     'webp' => 'image/webp',
40 ];
41
42 if (!isset($mimes[$ext])) { http_response_code(403); exit; }
43
44 header('Content-Type: ' . $mimes[$ext]);
45 header('Content-Length: ' . filesize($realFull));
46 header('Cache-Control: public, max-age=86400');
47 readfile($realFull);
48 exit;
```

51. ArchImp/public/index.php

```
1  <?php
2
3  require_once '../config/config.php';
4
5  $action = $_GET['action'] ?? 'login';
6
7  switch ($action) {
8      case 'register':
9          require_once BASE_PATH . '/controllers/AuthController.php';
10         $auth = new AuthController();
11         $auth->register();
12         break;
13
14     case 'login':
15         require_once BASE_PATH . '/controllers/AuthController.php';
16         $auth = new AuthController();
17         $auth->login();
18         break;
19
20     case 'logout':
21         require_once BASE_PATH . '/controllers/AuthController.php';
22         $auth = new AuthController();
23         $auth->logout();
24         break;
25
26     case 'forgot':
27         require_once BASE_PATH . '/controllers/AuthController.php';
28         $auth = new AuthController();
29         $auth->forgotPassword();
30         break;
31
32     case 'reset':
33         require_once BASE_PATH . '/controllers/AuthController.php';
34         $auth = new AuthController();
35         $auth->resetPassword();
36         break;
37
38     case 'dashboard':
39         require_once BASE_PATH . '/controllers/DashboardController.php';
40         $dash = new DashboardController();
41         $dash->index();
42         break;
43
44     case 'users':
45         require_once BASE_PATH . '/controllers/UserController.php';
46         $userCtrl = new UserController();
47         $userCtrl->index();
48         break;
49
50     case 'assign_role':
51         require_once BASE_PATH . '/controllers/UserController.php';
52         $userCtrl = new UserController();
53         $userCtrl->assignRole();
54         break;
55
56     case 'attendance_register':
57         require_once BASE_PATH . '/controllers/AttendanceController.php';
58         $attCtrl = new AttendanceController();
```

```
59         $attCtrl->register();
60         break;
61
62     case 'get_students':
63         require_once BASE_PATH . '/controllers/AttendanceController.php';
64         $attCtrl = new AttendanceController();
65         $attCtrl->getStudents();
66         break;
67
68     case 'attendance_view':
69         require_once BASE_PATH . '/controllers/AttendanceController.php';
70         $attCtrl = new AttendanceController();
71         $attCtrl->view();
72         break;
73
74     case 'my_attendance':
75         require_once BASE_PATH . '/controllers/AttendanceController.php';
76         $attCtrl = new AttendanceController();
77         $attCtrl->myAttendance();
78         break;
79
80     case 'academic':
81         require_once BASE_PATH . '/controllers/AcademicController.php';
82         $acadCtrl = new AcademicController();
83         $acadCtrl->index();
84         break;
85
86     case 'create_course':
87         require_once BASE_PATH . '/controllers/AcademicController.php';
88         $acadCtrl = new AcademicController();
89         $acadCtrl->createCourse();
90         break;
91
92     case 'create_subject':
93         require_once BASE_PATH . '/controllers/AcademicController.php';
94         $acadCtrl = new AcademicController();
95         $acadCtrl->createSubject();
96         break;
97
98     case 'enroll_students':
99         require_once BASE_PATH . '/controllers/AcademicController.php';
100        $acadCtrl = new AcademicController();
101        $acadCtrl->enrollStudents();
102        break;
103
104     case 'view_course_students':
105         require_once BASE_PATH . '/controllers/AcademicController.php';
106         $acadCtrl = new AcademicController();
107         $acadCtrl->viewCourseStudents();
108         break;
109
110     case 'course_subjects':
111         require_once BASE_PATH . '/controllers/AcademicController.php';
112         $acadCtrl = new AcademicController();
113         $acadCtrl->courseSubjects();
114         break;
115
116     case 'edit_course_subject':
117         require_once BASE_PATH . '/controllers/AcademicController.php';
```

```
118         $acadCtrl = new AcademicController();
119         $acadCtrl->editCourseSubject();
120         break;
121
122     case 'remove_course_subject':
123         require_once BASE_PATH . '/controllers/AcademicController.php';
124         $acadCtrl = new AcademicController();
125         $acadCtrl->removeCourseSubject();
126         break;
127
128     case 'assign_subject_teacher':
129         require_once BASE_PATH . '/controllers/AcademicController.php';
130         $acadCtrl = new AcademicController();
131         $acadCtrl->assignSubjectTeacher();
132         break;
133
134     case 'unassign_subject_teacher':
135         require_once BASE_PATH . '/controllers/AcademicController.php';
136         $acadCtrl = new AcademicController();
137         $acadCtrl->unassignSubjectTeacher();
138         break;
139
140     case 'reports':
141         require_once BASE_PATH . '/controllers/ReportController.php';
142         $reportCtrl = new ReportController();
143         $reportCtrl->index();
144         break;
145
146     case 'generate_pdf':
147         require_once BASE_PATH . '/controllers/ReportController.php';
148         $reportCtrl = new ReportController();
149         $reportCtrl->generatePDF();
150         break;
151
152     case 'generate_excel':
153         require_once BASE_PATH . '/controllers/ReportController.php';
154         $reportCtrl = new ReportController();
155         $reportCtrl->generateExcel();
156         break;
157
158     case 'manage_representatives':
159         require_once BASE_PATH . '/controllers/RepresentativeController.php';
160         $repCtrl = new RepresentativeController();
161         $repCtrl->manageRepresentatives();
162         break;
163
164     case 'my_children':
165         require_once BASE_PATH . '/controllers/RepresentativeController.php';
166         $repCtrl = new RepresentativeController();
167         $repCtrl->myChildren();
168         break;
169
170     case 'child_attendance':
171         require_once BASE_PATH . '/controllers/RepresentativeController.php';
172         $repCtrl = new RepresentativeController();
173         $repCtrl->childAttendance();
174         break;
175
176     case 'assignments':
```

```
177         require_once BASE_PATH . '/controllers/AssignmentController.php';
178         $assignCtrl = new AssignmentController();
179         $assignCtrl->index();
180         break;
181
182     case 'create_assignment':
183         require_once BASE_PATH . '/controllers/AssignmentController.php';
184         $assignCtrl = new AssignmentController();
185         $assignCtrl->assign();
186         break;
187
188     case 'set_tutor':
189         require_once BASE_PATH . '/controllers/AssignmentController.php';
190         $assignCtrl = new AssignmentController();
191         $assignCtrl->setTutor();
192         break;
193
194     case 'remove_tutor':
195         require_once BASE_PATH . '/controllers/AssignmentController.php';
196         $controller = new AssignmentController();
197         $controller->removeTutor();
198         break;
199
200     case 'get_course_teachers':
201         require_once BASE_PATH . '/controllers/AssignmentController.php';
202         $controller = new AssignmentController();
203         $controller->getCourseTeachers();
204         break;
205
206     case 'remove_assignment':
207         require_once BASE_PATH . '/controllers/AssignmentController.php';
208         $assignCtrl = new AssignmentController();
209         $assignCtrl->remove();
210         break;
211
212     case 'view_course_assignments':
213         require_once BASE_PATH . '/controllers/AssignmentController.php';
214         $assignCtrl = new AssignmentController();
215         $assignCtrl->viewByCourse();
216         break;
217
218     case 'profile':
219         require_once BASE_PATH . '/controllers/ProfileController.php';
220         $profileCtrl = new ProfileController();
221         $profileCtrl->view();
222         break;
223
224     case 'edit_profile':
225         require_once BASE_PATH . '/controllers/ProfileController.php';
226         $profileCtrl = new ProfileController();
227         $profileCtrl->edit();
228         break;
229
230     case 'change_password':
231         require_once BASE_PATH . '/controllers/ProfileController.php';
232         $profileCtrl = new ProfileController();
233         $profileCtrl->changePassword();
234         break;
235
```

```
236     case 'search_students':
237         require_once BASE_PATH . '/controllers/SearchController.php';
238         $searchCtrl = new SearchController();
239         $searchCtrl->searchStudents();
240         break;
241
242     case 'export_student_list':
243         require_once BASE_PATH . '/controllers/ExportController.php';
244         $exportCtrl = new ExportController();
245         $exportCtrl->exportStudentList();
246         break;
247
248     case 'stats':
249         require_once BASE_PATH . '/controllers/StatsController.php';
250         $statsCtrl = new StatsController();
251         $statsCtrl->index();
252         break;
253
254     case 'attendance_calendar':
255         require_once BASE_PATH . '/controllers/AttendanceController.php';
256         $attCtrl = new AttendanceController();
257         $attCtrl->calendar();
258         break;
259
260     case 'submit_justification':
261         require_once BASE_PATH . '/controllers/JustificationController.php';
262         $justCtrl = new JustificationController();
263         $justCtrl->submit();
264         break;
265
266     case 'my_justifications':
267         require_once BASE_PATH . '/controllers/JustificationController.php';
268         $justCtrl = new JustificationController();
269         $justCtrl->myJustifications();
270         break;
271
272     case 'pending_justifications':
273         require_once BASE_PATH . '/controllers/JustificationController.php';
274         $justCtrl = new JustificationController();
275         $justCtrl->pending();
276         break;
277
278     case 'reviewed_justifications':
279         require_once BASE_PATH . '/controllers/JustificationController.php';
280         $controller = new JustificationController();
281         $controller->reviewed();
282         break;
283
284     case 'review_justification':
285         require_once BASE_PATH . '/controllers/JustificationController.php';
286         $justCtrl = new JustificationController();
287         $justCtrl->review();
288         break;
289
290     case 'backups':
291         require_once BASE_PATH . '/controllers/BackupController.php';
292         $backupCtrl = new BackupController();
293         $backupCtrl->index();
294         break;
```



```
295
296     case 'create_backup':
297         require_once BASE_PATH . '/controllers/BackupController.php';
298         $backupCtrl = new BackupController();
299         $backupCtrl->create();
300         break;
301
302     case 'download_backup':
303         require_once BASE_PATH . '/controllers/BackupController.php';
304         $backupCtrl = new BackupController();
305         $backupCtrl->download();
306         break;
307
308     case 'cleanup_backups':
309         require_once BASE_PATH . '/controllers/BackupController.php';
310         $backupCtrl = new BackupController();
311         $backupCtrl->cleanup();
312         break;
313
314     case 'remove_role':
315         require_once BASE_PATH . '/controllers/UserController.php';
316         $userCtrl = new UserController();
317         $userCtrl->removeRole();
318         break;
319
320     case 'get_course_subjects':
321         require_once BASE_PATH . '/controllers/AttendanceController.php';
322         $controller = new AttendanceController();
323         $controller->getCourseSubjects();
324         break;
325
326     case 'get_teacher_course_subjects':
327         require_once BASE_PATH . '/controllers/AttendanceController.php';
328         $controller = new AttendanceController();
329         $controller->getTeacherCourseSubjects();
330         break;
331
332     case 'schedules':
333         require_once BASE_PATH . '/controllers/ScheduleController.php';
334         $controller = new ScheduleController();
335         $controller->index();
336         break;
337
338     case 'manage_schedule':
339         require_once BASE_PATH . '/controllers/ScheduleController.php';
340         $controller = new ScheduleController();
341         $controller->manageCourse();
342         break;
343
344     case 'delete_schedule_class':
345         require_once BASE_PATH . '/controllers/ScheduleController.php';
346         $controller = new ScheduleController();
347         $controller->deleteClass();
348         break;
349
350     case 'get_schedule_info':
351         require_once BASE_PATH . '/controllers/AttendanceController.php';
352         $controller = new AttendanceController();
353         $controller->getScheduleInfo();
```

```
354         break;
355
356     case 'get_existing_attendance':
357         require_once BASE_PATH . '/controllers/AttendanceController.php';
358         $controller = new AttendanceController();
359         $controller->getExistingAttendance();
360         break;
361
362     case 'tutor_management':
363         require_once BASE_PATH . '/controllers/AssignmentController.php';
364         $controller = new AssignmentController();
365         $controller->tutorManagement();
366         break;
367
368     case 'check_course_tutor':
369         require_once BASE_PATH . '/controllers/AssignmentController.php';
370         $controller = new AssignmentController();
371         $controller->checkCourseTutor();
372         break;
373
374     case 'get_course_subjects_schedule':
375         require_once BASE_PATH . '/controllers/ScheduleController.php';
376         $controller = new ScheduleController();
377         $controller->getCourseSubjectsSchedule();
378         break;
379
380     case 'check_schedule_conflict':
381         require_once BASE_PATH . '/controllers/ScheduleController.php';
382         $controller = new ScheduleController();
383         $controller->checkScheduleConflict();
384         break;
385
386     case 'institution':
387         require_once BASE_PATH . '/controllers/InstitutionController.php';
388         $controller = new InstitutionController();
389         $controller->index();
390         break;
391
392     case 'update_institution':
393         require_once BASE_PATH . '/controllers/InstitutionController.php';
394         $controller = new InstitutionController();
395         $controller->update();
396         break;
397
398     case 'assign_institution_shift':
399         require_once BASE_PATH . '/controllers/InstitutionController.php';
400         $controller = new InstitutionController();
401         $controller->assignShift();
402         break;
403
404     case 'remove_institution_shift':
405         require_once BASE_PATH . '/controllers/InstitutionController.php';
406         $controller = new InstitutionController();
407         $controller->removeShift();
408         break;
409
410     case 'toggle_institution_shift':
411         require_once BASE_PATH . '/controllers/InstitutionController.php';
412         (new InstitutionController())->toggleShift();
```

```
413         break;
414
415     case 'create_user':
416         require_once BASE_PATH . '/controllers/UserController.php';
417         $controller = new UserController();
418         $controller->create();
419         break;
420
421     case 'edit_user':
422         require_once BASE_PATH . '/controllers/UserController.php';
423         $controller = new UserController();
424         $controller->edit();
425         break;
426
427     case 'delete_user':
428         require_once BASE_PATH . '/controllers/UserController.php';
429         $controller = new UserController();
430         $controller->delete();
431         break;
432
433     case 'create_school_year':
434         require_once BASE_PATH . '/controllers/AcademicController.php';
435         $controller = new AcademicController();
436         $controller->createSchoolYear();
437         break;
438
439     case 'edit_school_year':
440         require_once BASE_PATH . '/controllers/AcademicController.php';
441         $controller = new AcademicController();
442         $controller->editSchoolYear();
443         break;
444
445     case 'delete_school_year':
446         require_once BASE_PATH . '/controllers/AcademicController.php';
447         $controller = new AcademicController();
448         $controller->deleteSchoolYear();
449         break;
450
451     case 'activate_school_year':
452         require_once BASE_PATH . '/controllers/AcademicController.php';
453         $controller = new AcademicController();
454         $controller->activateSchoolYear();
455         break;
456
457     case 'deactivate_school_year':
458         require_once BASE_PATH . '/controllers/AcademicController.php';
459         $controller = new AcademicController();
460         $controller->deactivateSchoolYear();
461         break;
462
463     case 'edit_course':
464         require_once BASE_PATH . '/controllers/AcademicController.php';
465         $controller = new AcademicController();
466         $controller->editCourse();
467         break;
468
469     case 'delete_course':
470         require_once BASE_PATH . '/controllers/AcademicController.php';
471         $controller = new AcademicController();
```

```
472         $controller->deleteCourse();
473         break;
474
475     case 'edit_subject':
476         require_once BASE_PATH . '/controllers/AcademicController.php';
477         $controller = new AcademicController();
478         $controller->editSubject();
479         break;
480
481     case 'delete_subject':
482         require_once BASE_PATH . '/controllers/AcademicController.php';
483         $controller = new AcademicController();
484         $controller->deleteSubject();
485         break;
486
487     case 'unenroll_student':
488         require_once BASE_PATH . '/controllers/AcademicController.php';
489         $controller = new AcademicController();
490         $controller->unenrollStudent();
491         break;
492
493     case 'delete_backup':
494         require_once BASE_PATH . '/controllers/BackupController.php';
495         $controller = new BackupController();
496         $controller->delete();
497         break;
498
499     case 'remove_representative':
500         require_once BASE_PATH . '/controllers/RepresentativeController.php';
501         $controller = new RepresentativeController();
502         $controller->removeRelation();
503         break;
504
505     case 'notifications':
506         require_once BASE_PATH . '/controllers/NotificationController.php';
507         (new NotificationController())->index();
508         break;
509
510     case 'notifications_mark_read':
511         require_once BASE_PATH . '/controllers/NotificationController.php';
512         (new NotificationController())->markRead();
513         break;
514
515     case 'notifications_mark_all':
516         require_once BASE_PATH . '/controllers/NotificationController.php';
517         (new NotificationController())->markAllRead();
518         break;
519
520     case 'notifications_delete':
521         require_once BASE_PATH . '/controllers/NotificationController.php';
522         (new NotificationController())->delete();
523         break;
524
525     case 'notifications_delete_read':
526         require_once BASE_PATH . '/controllers/NotificationController.php';
527         (new NotificationController())->deleteRead();
528         break;
529
530     case 'notifications_unread_json':
```

```
531         require_once BASE_PATH . '/controllers/NotificationController.php';
532         (new NotificationController())->getUnread();
533         break;
534
535     case 'tutor_course_attendance':
536         require_once BASE_PATH . '/controllers/TutorController.php';
537         (new TutorController())->courseAttendance();
538         break;
539
540     case 'tutor_course_attendance_ajax':
541         require_once BASE_PATH . '/controllers/TutorController.php';
542         (new TutorController())->ajax();
543         break;
544
545     default:
546         header('Location: ?action=login');
547 }
```

52. ArchImp/public/css/global.css

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

354     transition: border-color 0.2s;
355     font-family: inherit;
356 }
357 .form-control:focus {
358     border-color: #007bff;
359     outline: none;
360     box-shadow: 0 0 0 2px rgba(0,123,255,0.15);
361 }
362 .form-row {
363     display: grid;
364     grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));
365     gap: 16px;
366 }
367 .form-hint { font-size: 0.77rem; color: #888; margin-top: 3px; }
368
369 /* ■■■ Section divider ████████████████████████████████████████████ */
370 .section-title {
371     font-size: 0.82rem;
372     font-weight: 700;
373     color: #888;
374     text-transform: uppercase;
375     letter-spacing: 0.8px;
376     margin: 20px 0 10px;
377     padding-bottom: 6px;
378     border-bottom: 2px solid #e0e0e0;
379 }
380
381 /* ■■■ Action row (botonera superior de tablas) ████████████████████ */
382 .action-row {
383     display: flex;
384     justify-content: space-between;
385     align-items: center;
386     margin-bottom: 14px;
387     flex-wrap: wrap;
388     gap: 8px;
389 }
390 .action-row h2 {
391     font-size: 1rem;
392     font-weight: 700;
393     color: #333;
394 }
395
396 /* ■■■ Responsive ████████████████████████████████████████████ */
397 @media (max-width: 768px) {
398     .page-header { flex-direction: column; align-items: flex-start; gap: 10px; }
399     .page-header .ph-actions { margin-left: 0; }
400     .stats-row { flex-direction: column; }
401     .stat-card { min-width: unset; }
402     .filter-grid { grid-template-columns: 1fr 1fr; }
403     table { font-size: 0.8rem; }
404     thead th, tbody td { padding: 7px 8px; }
405 }
406 @media (max-width: 480px) {
407     .filter-grid { grid-template-columns: 1fr; }
408     .form-row { grid-template-columns: 1fr; }
409 }

```

53. ArchImp/views/academic/course_edit.php

```
1  <?php
2  require_once BASE_PATH . '/helpers/Security.php';
3  Security::requireLogin();
4  if (!Security::hasRole('autoridad')) { die('Acceso denegado'); }
5  ?>
6  <!DOCTYPE html>
7  <html lang="es">
8  <head>
9      <meta charset="UTF-8">
10     <title>Editar Curso - EcuAsist</title>
11     <style>
12         * { margin: 0; padding: 0; box-sizing: border-box; }
13         body { font-family: Arial, sans-serif; background: #f4f4f4; }
14         .container { max-width: 700px; margin: 30px auto; padding: 0 20px; }
15         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
16         .card-header { background: #007bff; color: white; padding: 15px 20px; border-radius: 8px 8px 0 0; margin: -30px -30px 0 -30px; }
17         .card-header h4 { margin: 0; font-size: 18px; }
18         .form-group { margin-bottom: 15px; }
19         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
20         input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; font-size: 14px; }
21         .btn { padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer; font-size: 14px; text-decoration: none; }
22         .btn-success { background: #28a745; color: white; }
23         .btn-secondary { background: #6c757d; color: white; }
24         .alert-danger { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 15px; }
25         .d-flex { display: flex; justify-content: space-between; margin-top: 20px; }
26     </style>
27 </head>
28 <body>
29     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
30
31     <div class="container">
32         <div class="card">
33             <div class="card-header">
34                 <h4>■ Editar Curso: <?= htmlspecialchars($course['name']) ?></h4>
35             </div>
36
37             <?php if (!empty($errors)): ?>
38                 <div class="alert-danger">
39                     <?php foreach ($errors as $error): ?>
40                         <div><?= htmlspecialchars($error) ?></div>
41                     <?php endforeach; ?>
42                 </div>
43             <?php endif; ?>
44
45             <form method="POST">
46
47                 <div class="form-group">
48                     <label>Nivel Educativo</label>
49                     <select name="education_type" id="education_type" required onchange="updateGradeLevels()">
50                         <option value="">Seleccionar tipo...</option>
51                         <option value="inicial">■ Educación Inicial</option>
52                         <option value="egb">■ Educación General Básica (EGB)</option>
53                         <option value="bgu">■ Bachillerato General Unificado (BGU)</option>
54                         <option value="bt">■ Bachillerato Técnico (BT)</option>
55                     </select>
56                 </div>
57
58                 <div class="form-group">
```

```

59         <label>Grado / Año</label>
60         <select name="grade_level" id="grade_level" required onchange="onGradeChange()">
61             <option value="">Seleccione nivel primero...</option>
62         </select>
63     </div>
64
65     <div class="form-group" id="group_specialty" style="display:none;">
66         <label>Figura Profesional</label>
67         <select name="specialty" id="specialty" onchange="updateCarreras()">
68             <option value="">Seleccionar figura...</option>
69             <option value="Informática">Informática</option>
70             <option value="Administración">Administración</option>
71             <option value="Contabilidad">Contabilidad</option>
72             <option value="Comercialización y Ventas">Comercialización y Ventas</option>
73             <option value="Servicios Hoteleros">Servicios Hoteleros</option>
74             <option value="Turismo">Turismo</option>
75             <option value="Electromecánica Automotriz">Electromecánica Automotriz</option>
76             <option value="Instalaciones Eléctricas">Instalaciones Eléctricas</option>
77             <option value="Electrónica de Consumo">Electrónica de Consumo</option>
78             <option value="Mecanizado y Construcciones Metálicas">Mecanizado y Construcciones Metálica.
79             <option value="Producción Agropecuaria">Producción Agropecuaria</option>
80             <option value="Acuicultura">Acuicultura</option>
81             <option value="Industrialización de Alimentos">Industrialización de Alimentos</option>
82             <option value="Confección Textil">Confección Textil</option>
83             <option value="Música">Música</option>
84             <option value="Artes Plásticas">Artes Plásticas</option>
85             <option value="Diseño Gráfico">Diseño Gráfico</option>
86             <option value="Servicios de Belleza">Servicios de Belleza</option>
87             <option value="Atención Integral en Salud">Atención Integral en Salud</option>
88             <option value="Mecatrónica">Mecatrónica</option>
89             <option value="Refrigeración y Climatización">Refrigeración y Climatización</option>
90             <option value="Construcción">Construcción</option>
91             <option value="Redes y Telecomunicaciones">Redes y Telecomunicaciones</option>
92             <option value="Seguridad Industrial">Seguridad Industrial</option>
93             <option value="Logística y Transporte">Logística y Transporte</option>
94         </select>
95     </div>
96
97     <div class="form-group" id="group_carrera" style="display:none;">
98         <label>Especialidad / Carrera <span style="color:#999; font-weight:normal;">(opcional)&
99         <select name="carrera" id="carrera" onchange="generateCourseName()">
100             <option value="">Sin especificar</option>
101         </select>
102     </div>
103
104     <div class="form-group">
105         <label>Paralelo</label>
106         <select name="parallel" id="parallel" required onchange="generateCourseName()">
107             <option value="">Seleccionar...</option>
108             <?php foreach(range('A', 'J') as $letter):>
109                 <option value="<? $letter ?>" <? $course['parallel'] == $letter ? 'selected' :
110                     <? $letter ?>
111                 </option>
112             <?php endforeach;>
113         </select>
114     </div>
115
116     <div class="form-group">
117         <label>Jornada</label>

```

```

118         <select name="shift_id" id="shift_id" required onchange="generateCourseName()">
119             <option value="">Seleccionar...</option>
120             <?php foreach($shifts as $shift): ?>
121                 <option value="<?> $shift['id'] ?>"
122                     data-shift="<?> $shift['name'] ?>"
123                     <?> $course['shift_id'] == $shift['id'] ? 'selected' : '' ?>>
124                     <?> ucfirst($shift['name']) ?>
125                 </option>
126             <?php endforeach; ?>
127         </select>
128     </div>
129
130     <div class="form-group">
131         <label>Nombre del Curso <span style="color:#999; font-weight:normal;">(generado automáti.
132         <input type="text" name="name" id="course_name" readonly
133             style="background: #f0f0f0; font-weight:bold;"
134             value="<?> htmlspecialchars($course['name']) ?>">
135     </div>
136
137     <div class="d-flex">
138         <a href="?action=academic" class="btn btn-secondary">← Cancelar</a>
139         <button type="submit" class="btn btn-success">✓ Guardar Cambios</button>
140     </div>
141 </form>
142 </div>
143 </div>
144
145 <script>
146 // Datos del curso actual para precargar
147 const currentGradeLevel = <?> json_encode($course['grade_level']) ?>;
148 const currentShiftId     = <?> json_encode((string)$course['shift_id']) ?>;
149
150 const gradeLevels = {
151     inicial: ['Inicial 1 (0-3 años)', 'Inicial 2 (3-5 años)',
152     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', '.
153     bgu: ['1.º BGU', '2.º BGU', '3.º BGU'],
154     bt: ['1.º BT', '2.º BT', '3.º BT']
155 };
156
157 const egbNocturnaAllowed = ['8.º EGB', '9.º EGB', '10.º EGB'];
158
159 const carreras = {
160     'Informática': ['Aplicaciones Informáticas', 'Programación de Software', 'Soporte Técnico', 'Sistemas Microinformát.
161     'Administración': ['Asistencia Administrativa', 'Gestión Empresarial'],
162     'Contabilidad': ['Contabilidad', 'Ventas e Información Comercial'],
163     'Comercialización y Ventas': ['Ventas e Información Comercial', 'Marketing'],
164     'Servicios Hoteleros': ['Hotelería', 'Hospitalidad'],
165     'Turismo': ['Turismo', 'Guía Turístico'],
166     'Electromecánica Automotriz': ['Electromecánica Automotriz', 'Mecánica Automotriz'],
167     'Instalaciones Eléctricas': ['Electricidad', 'Instalaciones Eléctricas'],
168     'Electrónica de Consumo': ['Electrónica', 'Mantenimiento Electrónico'],
169     'Atención Integral en Salud': ['Atención en Enfermería', 'Auxiliar de Salud'],
170     'Producción Agropecuaria': ['Producción Agropecuaria', 'Agroindustria'],
171     'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
172     'Diseño Gráfico': ['Diseño Gráfico', 'Multimedia'],
173     'Servicios de Belleza': ['Peluquería', 'Cosmetología']
174 };
175
176 // Detectar tipo educativo desde el grade_level actual

```

```

177     function detectType(grade) {
178         if (!grade) return '';
179         if (grade.includes('Inicial')) return 'inicial';
180         if (grade.includes('EGB')) return 'egb';
181         if (grade.includes('BGU')) return 'bgu';
182         if (grade.includes('BT')) return 'bt';
183         return '';
184     }
185
186     function getNocturnaOption() {
187         const opts = document.getElementById('shift_id').options;
188         for (let i = 0; i < opts.length; i++) {
189             if ((opts[i].getAttribute('data-shift') || '').toLowerCase() === 'nocturna') return opts[i];
190         }
191         return null;
192     }
193
194     function updateNocturnaVisibility(type, grade) {
195         const nocturna = getNocturnaOption();
196         if (!nocturna) return;
197         let visible = (type === 'bgu' || type === 'bt' || (type === 'egb' && egbNocturnaAllowed.includes(grade)));
198         nocturna.style.display = visible ? '' : 'none';
199         if (!visible && nocturna.selected) {
200             document.getElementById('shift_id').value = '';
201         }
202     }
203
204     function updateGradeLevels(preselectGrade) {
205         const type = document.getElementById('education_type').value;
206         const gradeSelect = document.getElementById('grade_level');
207         gradeSelect.innerHTML = '<option value="">Seleccionar grado...</option>';
208
209         if (!type) return;
210
211         (gradeLevels[type] || []).forEach(function(g) {
212             const opt = document.createElement('option');
213             opt.value = g;
214             opt.textContent = g;
215             if (preselectGrade && g === preselectGrade) opt.selected = true;
216             gradeSelect.appendChild(opt);
217         });
218
219         const isBT = (type === 'bt');
220         document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
221         document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
222         document.getElementById('specialty').required = isBT;
223
224         if (!isBT) {
225             document.getElementById('specialty').value = '';
226             document.getElementById('carrera').innerHTML = '<option value="">Sin especificar</option>';
227         }
228
229         updateNocturnaVisibility(type, gradeSelect.value);
230     }
231
232     function onGradeChange() {
233         const type = document.getElementById('education_type').value;
234         const grade = document.getElementById('grade_level').value;
235         updateNocturnaVisibility(type, grade);

```



```

236     generateCourseName();
237 }
238
239 function updateCarreras() {
240     const fig = document.getElementById('specialty').value;
241     const carreraSelect = document.getElementById('carrera');
242     carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
243     if (carreras[fig]) {
244         carreras[fig].forEach(function(c) {
245             const opt = document.createElement('option');
246             opt.value = c; opt.textContent = c;
247             carreraSelect.appendChild(opt);
248         });
249     }
250     generateCourseName();
251 }
252
253 function generateCourseName() {
254     const grade = document.getElementById('grade_level').value;
255     const parallel = document.getElementById('parallel').value;
256     const shiftSelect = document.getElementById('shift_id');
257     const shiftOpt = shiftSelect.options[shiftSelect.selectedIndex];
258     const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';
259     const type = document.getElementById('education_type').value;
260     const specialty = document.getElementById('specialty').value;
261     const carrera = document.getElementById('carrera').value;
262
263     if (!grade || !parallel || !shiftName) return;
264
265     let name = grade + ' ' + parallel + ' ';
266     if (type === 'bt' && specialty) {
267         name += ' - ' + specialty;
268         if (carrera) name += ' (' + carrera + ')';
269     }
270     name += ' - ' + shiftName.charAt(0).toUpperCase() + shiftName.slice(1);
271     document.getElementById('course_name').value = name;
272 }
273
274 // Extraer figura profesional y carrera desde el nombre del curso
275 // Formato: "1.º BT "A" - Informática (Soporte Técnico) - Matutina"
276 function parseSpecialtyFromName(courseName) {
277     const result = { specialty: '', carrera: '' };
278     // Buscar patrón: - Figura (Carrera) - o - Figura -
279     const match = courseName.match(/BT "[A-Z]" - ([^()]+)?(?:\s*\([^\)]+\))?\s*-\s*\w+$/);
280     if (match) {
281         result.specialty = match[1].trim();
282         result.carrera = match[2] ? match[2].trim() : '';
283     }
284     return result;
285 }
286
287 // PRECARGAR datos del curso al cargar la página
288 window.addEventListener('DOMContentLoaded', function() {
289     const detectedType = detectType(currentGradeLevel);
290     if (detectedType) {
291         document.getElementById('education_type').value = detectedType;
292         updateGradeLevels(currentGradeLevel);
293     }
294 }

```

```

295 // Preseleccionar jornada
296 const shiftSelect = document.getElementById('shift_id');
297 for (let i = 0; i < shiftSelect.options.length; i++) {
298     if (shiftSelect.options[i].value === currentShiftId) {
299         shiftSelect.selectedIndex = i;
300         break;
301     }
302 }
303
304 // Precargar figura profesional y carrera (solo BT)
305 if (detectedType === 'bt') {
306     const parsed = parseSpecialtyFromName(currentGradeLevel + ' - ' + &lt;? json_encode($course['name']) ?>);
307
308     // Intentar parsear directo del nombre completo del curso
309     const fullName = &lt;? json_encode($course['name']) ?>;
310     const matchFull = fullName.match(/- ([^()]+)(?:\s*\(((^)+)\))?\s*-\s*\w+$/);
311     if (matchFull) {
312         const fig = matchFull[1].trim();
313         const car = matchFull[2] ? matchFull[2].trim() : '';
314
315         // Seleccionar figura profesional
316         const specialtySelect = document.getElementById('specialty');
317         for (let i = 0; i < specialtySelect.options.length; i++) {
318             if (specialtySelect.options[i].value === fig) {
319                 specialtySelect.selectedIndex = i;
320                 break;
321             }
322         }
323
324         // Cargar carreras de esa figura y preseleccionar
325         if (fig & & carreras[fig]) {
326             const carreraSelect = document.getElementById('carrera');
327             carreraSelect.innerHTML = '&lt;option value=""&gt;Sin especificar&lt;/option&gt;';
328             carreras[fig].forEach(function(c) {
329                 const opt = document.createElement('option');
330                 opt.value = c; opt.textContent = c;
331                 if (c === car) opt.selected = true;
332                 carreraSelect.appendChild(opt);
333             });
334         }
335     }
336 }
337
338 updateNocturnaVisibility(detectedType, currentGradeLevel);
339 });
340 &lt;/script&gt;
341 &lt;/body&gt;
342 &lt;/html&gt;

```

54. ArchImp/views/academic/course_students.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Estudiantes del Curso - EcuAsist</title>
8      <style>
9          * { margin: 0; padding: 0; box-sizing: border-box; }
10         body { font-family: Arial, sans-serif; background: #f4f4f4; }
11         .container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
13         table { width: 100%; border-collapse: collapse; margin-top: 20px; }
14         th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
15         th { background: #f8f9fa; font-weight: bold; }
16         h2 { margin-bottom: 10px; color: #333; }
17         .course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
18         .btn-back { display: inline-block; margin-bottom: 20px; padding: 8px 16px; background: #6c757d; color: white; text-decoration: none; }
19         .btn-back:hover { background: #5a6268; }
20     </style>
21 </head>
22 <body>
23
24 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
25
26 <div class="breadcrumb">
27     ■ Inicio &rsquo; <a href="?action=academic">Configuración Académica</a> &rsquo; Estudiantes de
28 </div>
29
30 <div class="container">
31     <a href="?action=academic" class="btn-back">← Volver a Configuración Académica</a>
32
33     <div class="card">
34         <div class="course-info">
35             <h2><?php echo $course['name']; ?></h2>
36             <p><strong>Nivel:</strong> <?php echo $course['grade_level']; ?> | <strong>Paralelo:</strong> <?php echo $course['shift_name']; ?>
37             <p><strong>Jornada:</strong> <?php echo ucfirst($course['shift_name']); ?> | <strong>Año:</strong> <?php echo $course['year']; ?></p>
38         </div>
39
40         <h3>Estudiantes Matriculados (<?php echo count($students); ?>)</h3>
41
42         <?php if(count($students) > 0): ?>
43             <table>
44                 <thead>
45                     <tr>
46                         <th>#</th>
47                         <th>Apellidos y Nombres</th>
48                         <th>Cédula</th>
49                         <th>Fecha Matrícula</th>
50                     </tr>
51                 </thead>
52                 <tbody>
53                     <?php foreach($students as $index => $student): ?>
54                         <tr>
55                             <td><?php echo $index + 1; ?></td>
56                             <td><?php echo $student['last_name'] . ' ' . $student['first_name']; ?></td>
57                             <td><?php echo $student['dni']; ?></td>
58                             <td><?php echo date('d/m/Y', strtotime($student['enrollment_date'])); ?></td>
```

```
59         &lt;/tr&gt;
60         &lt;?php endforeach; ?&gt;
61     &lt;/tbody&gt;
62 &lt;/table&gt;
63 &lt;?php else: ?&gt;
64     &lt;p style="color: #666; margin-top: 20px;"&gt;No hay estudiantes matriculados en este curso.&lt;/p&gt;
65 &lt;?php endif; ?&gt;
66 &lt;/div&gt;
67 &lt;/div&gt;
68
69 &lt;/body&gt;
70 &lt;/html&gt;
```

55. ArchImp/views/academic/course_subject_edit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Editor Asignatura - EcuAsist</title>
8      <style>
9          * { margin: 0; padding: 0; box-sizing: border-box; }
10         body { font-family: Arial, sans-serif; background: #f4f4f4; }
11         .container { max-width: 500px; margin: 40px auto; padding: 0 20px; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
13         .card h2 { margin-bottom: 24px; color: #333; }
14         .form-group { margin-bottom: 18px; }
15         label { display: block; margin-bottom: 6px; font-weight: bold; color: #555; font-size: 14px; }
16         input { width: 100%; padding: 10px 12px; border: 1px solid #ddd; border-radius: 4px; font-size: 14px; }
17         input:focus { border-color: #007bff; outline: none; box-shadow: 0 0 0 2px rgba(0,123,255,.2); }
18         .btn-row { display: flex; gap: 10px; margin-top: 24px; }
19         .btn { padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer; font-size: 14px; text-decoration: none; }
20         .btn-save { background: #28a745; color: white; }
21         .btn-back { background: #6c757d; color: white; }
22         .btn:hover { opacity: 0.88; }
23     </style>
24 </head>
25 <body>
26
27 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
28
29 <div class="breadcrumb">
30     ■ Inicio &rsquo;
31     <a href="?action=academic">Configuración Académica</a> &rsquo;
32     <a href="?action=course_subjects&course_id=<?php echo $courseId ?>">Asignaturas del Curso</a> &rsquo;
33     Editor Asignatura
34 </div>
35
36 <div class="container">
37     <div class="card">
38         <h2>■ Editor Asignatura</h2>
39
40         <form method="POST" action="?action=edit_course_subject&subject_id=<?php echo $subjectId ?>&course_id=<?php echo $courseId ?>">
41             <div class="form-group">
42                 <label>Nombre de la asignatura *</label>
43                 <input type="text" name="name"
44                     value="<?php echo htmlspecialchars($subject['name']) ?>"
45                     required autofocus>
46             </div>
47             <div class="form-group">
48                 <label>Código <span style="font-weight:normal;color:#999;">(opcional)</span></label>
49                 <input type="text" name="code"
50                     value="<?php echo htmlspecialchars($subject['code']) ?>"
51                     placeholder="Ej: MAT">
52             </div>
53             <div class="btn-row">
54                 <button type="submit" class="btn btn-save">■ Guardar cambios</button>
55                 <a href="?action=course_subjects&course_id=<?php echo $courseId ?>" class="btn btn-back">← CANCELAR</a>
56             </div>
57         </form>
58     </div>
59 </div>
```

```
59 </div>
60
61 </body>
62 </html>
```

56. ArchImp/views/academic/course_subjects.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Asignaturas del Curso - EcuAsist</title>
8      <style>
9          * { margin: 0; padding: 0; box-sizing: border-box; }
10         body { font-family: Arial, sans-serif; background: #f4f4f4; }
11         .container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot.
13         .course-header { background: linear-gradient(135deg, #007bff, #0056b3); color: white; padding: 20px 30px; border.
14         .course-header h2 { font-size: 22px; margin-bottom: 6px; }
15         .course-header p { font-size: 14px; opacity: 0.9; }
16         table { width: 100%; border-collapse: collapse; margin-top: 10px; }
17         th, td { padding: 11px 13px; text-align: left; border-bottom: 1px solid #eee; font-size: 14px; }
18         th { background: #f8f9fa; font-weight: bold; color: #555; }
19         tr:hover td { background: #f0f7ff; }
20         .btn { display: inline-block; padding: 5px 12px; border-radius: 4px; border: none; cursor: pointer; font-size: 1.
21         .btn-edit { background: #ffc107; color: #212529; }
22         .btn-delete { background: #dc3545; color: white; }
23         .btn-back { background: #6c757d; color: white; padding: 8px 18px; font-size: 14px; }
24         .btn-add { background: #28a745; color: white; padding: 10px 20px; font-size: 14px; }
25         .btn-assign { background: #e65100; color: white; }
26         .btn-unassign { background: #6c757d; color: white; }
27         .btn:hover { opacity: 0.88; }
28         .alert { padding: 12px 16px; border-radius: 4px; margin-bottom: 16px; font-size: 14px; }
29         .alert-success { background: #d4edda; color: #155724; }
30         .alert-error { background: #f8d7da; color: #721c24; }
31         .form-row { display: flex; gap: 12px; align-items: flex-end; flex-wrap: wrap; }
32         .form-group { display: flex; flex-direction: column; flex: 1; min-width: 180px; }
33         .form-group label { font-size: 13px; font-weight: bold; color: #555; margin-bottom: 5px; }
34         .form-group input, .form-group select { padding: 9px 12px; border: 1px solid #ddd; border-radius: 4px; font-size.
35         .badge-count { background: #007bff; color: white; border-radius: 12px; padding: 2px 10px; font-size: 13px; margi.
36         .empty-msg { text-align: center; color: #999; padding: 30px; }
37         .teacher-ok { color: #2e7d32; font-weight: bold; font-size: 13px; }
38         .teacher-no { color: #bbb; font-style: italic; font-size: 13px; }
39         .modal-overlay { display:none; position:fixed; top:0; left:0; width:100%; height:100%; background:rgba(0,0,0,.5).
40         .modal-overlay.active { display:flex; }
41         .modal-box { background:white; padding:28px; border-radius:8px; max-width:420px; width:90%; box-shadow:0 4px 20p.
42         .modal-box h3 { margin-bottom:14px; color:#e65100; }
43         .modal-box select { width:100%; padding:10px; border:1px solid #ddd; border-radius:4px; font-size:14px; margin-b.
44         .modal-btns { display:flex; gap:10px; justify-content:flex-end; }
45         .modal-btns button { padding:8px 18px; border:none; border-radius:4px; cursor:pointer; font-size:14px; }
46     </style>
47 </head>
48 <body>
49
50 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
51
52 <div class="breadcrumb">
53     ■ Inicio &rsquo; <a href="?action=academic">Configuración Académica</a> &rsquo; Asignaturas de.
54 </div>
55
56 <div class="container">
57
58     <div class="course-header">
```

```

59         <h2>■ <?php echo htmlspecialchars($course['name']) ?></h2>
60     <p>
61         <strong>Nivel:</strong> <?php echo htmlspecialchars($course['grade_level']) ?> &nbsp;|&nbsp;
62         <strong>Paralelo:</strong> <?php echo htmlspecialchars($course['parallel']) ?> &nbsp;|&nbsp;
63         <strong>Jornada:</strong> <?php echo ucfirst($course['shift_name']) ?> &nbsp;|&nbsp;
64         <strong>Año Lectivo:</strong> <?php echo htmlspecialchars($course['year_name']) ?>
65     </p>
66 </div>
67
68 <?php if (isset($_GET['added'])): ?><div class="alert alert-success">✓ Asignatura agregada correcta.
69 <?php if (isset($_GET['updated'])): ?><div class="alert alert-success">✓ Asignatura actualizada.</div>
70 <?php if (isset($_GET['removed'])): ?><div class="alert alert-success">✓ Asignatura quitada del curso.
71 <?php if (isset($_GET['assigned'])): ?><div class="alert alert-success">✓ Docente asignado correctamen.
72 <?php if (isset($_GET['unassigned'])): ?><div class="alert alert-success">✓ Docente removido.</div>
73 <?php if (isset($_GET['error']) & $_GET['error'] === 'has_teacher'): ?>
74     <div class="alert alert-error">✗ No se puede quitar: esta asignatura tiene un docente asignado.</div>
75 <?php endif; ?>
76 <?php if (isset($_GET['assign_error'])): ?>
77     <div class="alert alert-error">✗ <?php echo htmlspecialchars($_GET['assign_error']) ?></div>
78 <?php endif; ?>
79
80 <div class="card">
81     <h3 style="margin-bottom:16px; color:#333;">
82         Asignaturas y Docentes
83     <span class="badge-count"><?php echo count($subjects) ?></span>
84 </h3>
85
86     <?php if (empty($subjects)): ?>
87         <div class="empty-msg">■ Este curso aún no tiene asignaturas asignadas.</div>
88     <?php else: ?>
89         <table>
90             <thead>
91                 <tr>
92                     <th>#</th>
93                     <th>Asignatura</th>
94                     <th>Código</th>
95                     <th>Docente</th>
96                     <th style="width:220px; text-align:center;">Acciones</th>
97                 </tr>
98             </thead>
99             <tbody>
100                 <?php foreach ($subjects as $i => $sub): ?>
101                     <tr>
102                         <td><?php echo $i + 1 ?></td>
103                         <td><strong><?php echo htmlspecialchars($sub['name']) ?></strong></td>
104                         <td><?php echo htmlspecialchars($sub['code']) ?>: '<span style="color:#bbb;">-</span>'.
105                         <td>
106                             <?php if (!empty($sub['teacher_name'])): ?>
107                                 <span class="teacher-ok">■ <?php echo htmlspecialchars($sub['teacher_name']) ?></span>
108                             <?php else: ?>
109                                 <span class="teacher-no">Sin docente</span>
110                             <?php endif; ?>
111                         </td>
112                         <td style="text-align:center; white-space:nowrap;">
113                             <button class="btn btn-assign"
114                                 onclick="abrirModal(<?php echo $sub['id'] ?>, '<?php echo addslashes($sub['name']) ?>
115                                 <?php echo empty($sub['teacher_name']) ? '■ Asignar' : '■ Cambiar' ?>
116                             </button>
117

```



```

118         <?php if (!empty($sub['teacher_name'])): ?>
119             <form method="POST" action="?action=unassign_subject_teacher" style="display:inline;">
120                 <input type="hidden" name="assignment_id" value="<? $sub['assignment_id'] ?> ">
121                 <input type="hidden" name="course_id" value="<? $course['id'] ?> ">
122                 <button type="submit" class="btn btn-unassign"
123                     onclick="return confirm('¿Quitar docente de esta asignatura?') ">✕</button>
124             </form>
125         <?php endif; ?>
126
127         <a href="?action=edit_course_subject&subject_id=<? $sub['id'] ?> &course_id=&
128             class="btn btn-edit">✎</a>
129
130         <form method="POST" action="?action=remove_course_subject" style="display:inline;"
131             onsubmit="return confirmRemove(event, ' <? htmlspecialchars(addslashes($sub['name'])) ?."
132             <input type="hidden" name="course_id" value="<? $course['id'] ?> ">
133             <input type="hidden" name="subject_id" value="<? $sub['id'] ?> ">
134             <button type="submit" class="btn btn-delete">✖</button>
135         </form>
136     </td>
137 </tr>
138 <?php endforeach; ?>
139 </tbody>
140 </table>
141 <?php endif; ?>
142 </div>
143
144 <div class="card">
145     <h3 style="margin-bottom:16px; color:#333;">✚ Agregar asignatura a este curso</h3>
146     <form method="POST" action="?action=course_subjects&course_id=<? $course['id'] ?> ">
147         <div class="form-row">
148             <div class="form-group">
149                 <label>Nombre </label>
150                 <input type="text" name="new_subject_name" placeholder="Ej: Emprendimiento" required>
151             </div>
152             <div class="form-group" style="max-width:160px;">
153                 <label>Código <span style="font-weight:normal;color:#999;">(opcional)</span> </div>
154                 <input type="text" name="new_subject_code" placeholder="Ej: EMP">
155             </div>
156             <button type="submit" class="btn btn-add">✚ Agregar</button>
157         </div>
158     </form>
159 </div>
160
161     <a href="?action=academic" class="btn btn-back">← Volver a Configuración Académica</a>
162 </div>
163
164 <!-- Modal: Asignar Docente -->
165 <div class="modal-overlay" id="modalAsignar">
166     <div class="modal-box">
167         <h3>✚ Asignar Docente</h3>
168         <p id="modalSubjectLabel" style="font-size:15px; color:#333; margin-bottom:14px;">
169             <form method="POST" action="?action=assign_subject_teacher">
170                 <input type="hidden" name="course_id" value="<? $course['id'] ?> ">
171                 <input type="hidden" name="subject_id" id="modalSubjectId">
172                 <input type="hidden" name="assignment_id" id="modalAssignmentId">
173                 <label style="font-size:13px; font-weight:bold; color:#555; display:block; margin-bottom:6px;">Docente.
174                 <select name="teacher_id" id="modalTeacherSelect" required>
175                     <option value="">Seleccionar docente...</option>
176                 <?php foreach ($teachers as $t): ?>

```

```

177         <option value="&lt;?= ${'id'} ?&gt;"&gt;
178             &lt;?= htmlspecialchars("${'last_name'} . ' ' . ${'first_name'})" ?&gt;
179         </option>
180     <?php endforeach; ?&gt;
181 </select>
182 <div class="modal-btns"&gt;
183     <button type="button" onclick="cerrarModal()" style="background:#6c757d;color:white;"&gt;Cancelar&lt;/button>
184     <button type="submit" style="background:#e65100;color:white;"&gt;✓ Guardar&lt;/button>
185 </div>
186 </form>
187 </div>
188 </div>
189
190 <script>
191 function abrirModal(subjectId, subjectName, assignmentId, teacherId) {
192     document.getElementById('modalSubjectId').value = subjectId;
193     document.getElementById('modalSubjectLabel').textContent = '■ ' + subjectName;
194     document.getElementById('modalAssignmentId').value = assignmentId ?? '';
195     // Pre-seleccionar docente actual si existe
196     const sel = document.getElementById('modalTeacherSelect');
197     sel.value = teacherId ?? '';
198     document.getElementById('modalAsignar').classList.add('active');
199 }
200 function cerrarModal() {
201     document.getElementById('modalAsignar').classList.remove('active');
202 }
203 document.getElementById('modalAsignar').addEventListener('click', function(e) {
204     if (e.target === this) cerrarModal();
205 });
206 function confirmRemove(event, nombre) {
207     event.preventDefault();
208     const m = document.createElement('div');
209     m.style.cssText = 'position:fixed;top:0;left:0;width:100%;height:100%;background:rgba(0,0,0,.5);display:flex;align-items:center;justify-content:center;';
210     m.innerHTML = `<div style="background:white;padding:28px;border-radius:8px;max-width:420px;width:90%;box-shadow:0 0 10px #000;">
211         <h3 style="color:#dc3545;margin-bottom:12px;"&gt;■ ■ ■ Quitar asignatura&lt;/h3>
212         <p style="color:#555;margin-bottom:18px;"&gt;¿Quitar <strong>${nombre}</strong> de este curso?&lt;/p>
213         <div style="display:flex;gap:10px;justify-content:flex-end;"&gt;
214             <button id="bC" style="padding:8px 18px;background:#6c757d;color:white;border:none;border-radius:4px;cursor:pointer;"&gt;Cancelar
215             <button id="bO" style="padding:8px 18px;background:#dc3545;color:white;border:none;border-radius:4px;cursor:pointer;"&gt;Quitar
216         </div>
217     </div>`;
218     document.body.appendChild(m);
219     const form = event.target;
220     document.getElementById('bO').onclick = () => { document.body.removeChild(m); form.submit(); };
221     document.getElementById('bC').onclick = () => document.body.removeChild(m);
222     return false;
223 }
224 </script>
225 </body>
226 </html>

```

57. ArchImp/views/academic/create.php

[illegible]

```

59         </ul></div></div>
60     </div>
61 <?php endif; ?>
62
63 <div class="info-box">
64     <strong>■ Información importante:</strong>
65     <ul class="mb-0 mt-2">
66         <li>Solo puede haber un año lectivo activo a la vez</li>
67         <li>Las fechas no pueden solaparse con otros años lectivos</li>
68         <li>Si marcas como activo, los demás se desactivarán automáticamente</li>
69     </ul>
70 </div>
71
72 <form method="POST" action="?action=create_school_year">
73     <div class="form-section">
74         <h5>■ Información General</h5>
75
76         <div class="mb-3">
77             <label class="form-label required">Nombre del Año Lectivo</label>
78             <input type="text"
79                 name="name"
80                 class="form-control"
81                 placeholder="Ej: 2025-2026"
82                 value="<? isset($_POST['name']) ? htmlspecialchars($_POST['name']) : '' ?."
83                 required>
84             <small class="text-muted">Formato recomendado: YYYY-YYYY</small>
85         </div>
86
87         <div class="row">
88             <div class="col-md-6 mb-3">
89                 <label class="form-label required">Fecha de Inicio</label>
90                 <input type="date"
91                     name="start_date"
92                     class="form-control"
93                     value="<? isset($_POST['start_date']) ? $_POST['start_date'] : '' ?&
94                     required>
95             </div>
96             <div class="col-md-6 mb-3">
97                 <label class="form-label required">Fecha de Fin</label>
98                 <input type="date"
99                     name="end_date"
100                     class="form-control"
101                     value="<? isset($_POST['end_date']) ? $_POST['end_date'] : '' ?&
102                     required>
103             </div>
104         </div>
105
106         <div class="form-check">
107             <input class="form-check-input"
108                 type="checkbox"
109                 name="is_active"
110                 id="is_active"
111                 <? isset($_POST['is_active']) ? 'checked' : '' ?&
112             <label class="form-check-label" for="is_active">
113                 <strong>Activar este año lectivo</strong>
114                 <br>
115                 <small class="text-muted">
116                     Si se activa, todos los demás años lectivos se desactivarán automáticamente
117                 </small>

```

```
118         </label>;
119     </div>;
120 </div>;
121
122     <div class="d-flex justify-content-between mt-4">
123         <a href="?action=academic" class="btn btn-secondary">
124             ← Cancelar
125         </a>;
126         <button type="submit" class="btn btn-primary">
127             ✓ Crear Año Lectivo
128         </button>;
129     </div>;
130 </form>;
131 </div>;
132 </div>;
133 </div>;
134 </div>;
135 </div>;
136
137     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
138 </body>;
139 </html>;
```

58. ArchImp/views/academic/edit.php

```
1  <?php
2  require_once BASE_PATH . '/helpers/Security.php';
3  Security::requireLogin();
4  if (!Security::hasRole('autoridad')) {
5      die('Acceso denegado');
6  }
7  ?>
8  <!DOCTYPE html>
9  <html lang="es">
10 <head>
11     <meta charset="UTF-8">
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <title>Editar Año Lectivo - EcuAsist</title>
14     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
15     <style>
16         body { padding-top: 70px; }
17         .form-section {
18             background: #f8f9fa;
19             padding: 20px;
20             border-radius: 8px;
21             margin-bottom: 20px;
22         }
23         .form-section h5 {
24             color: #007bff;
25             border-bottom: 2px solid #007bff;
26             padding-bottom: 10px;
27             margin-bottom: 20px;
28         }
29         .required::after {
30             content: " *";
31             color: red;
32         }
33         .info-box {
34             background: #d1ecf1;
35             border-left: 4px solid #0c5460;
36             padding: 15px;
37             margin-bottom: 20px;
38         }
39         .status-badge {
40             display: inline-block;
41             padding: 8px 15px;
42             border-radius: 20px;
43             font-weight: bold;
44             font-size: 14px;
45         }
46         .status-active {
47             background: #28a745;
48             color: white;
49         }
50         .status-inactive {
51             background: #6c757d;
52             color: white;
53         }
54     </style>
55 </head>
56 <body>
57     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
58
```

```

59     <div class="container">
60         <div class="row justify-content-center">
61             <div class="col-md-8">
62                 <div class="card shadow">
63                     <div class="card-header bg-primary text-white">
64                         <div class="d-flex justify-content-between align-items-center">
65                             <h4 class="mb-0">■ Editar Año Lectivo: <?= htmlspecialchars($year['name']) ?> &
66                             <span class="status-badge <?= $year['is_active'] ? 'status-active' : 'status-inactive'.
67                                 <?= $year['is_active'] ? '✓ ACTIVO' : 'Inactivo' ?>
68                             </span>
69                         </div>
70                     </div>
71                     <div class="card-body">
72                         <?php if (!empty($errors)): ?>
73                             <div class="alert alert-danger">
74                                 <strong>Errores:</strong>
75                                 <ul class="mb-0">
76                                     <?php foreach ($errors as $error): ?>
77                                         <li><?= htmlspecialchars($error) ?></li>
78                                     <?php endforeach; ?>
79                                 </ul>
80                             </div>
81                         <?php endif; ?>
82
83                         <div class="info-box">
84                             <strong>■ Información importante:</strong>
85                             <ul class="mb-0 mt-2">
86                                 <li>Las fechas no pueden solaparse con otros años lectivos</li>
87                                 <li>El estado (activo/inactivo) se gestiona desde la tabla principal</li>
88                                 <li>No se puede eliminar un año lectivo activo</li>
89                             </ul>
90                         </div>
91
92                         <form method="POST" action="?action=edit_school_year&id=<?= $year['id'] ?> ">
93                             <div class="form-section">
94                                 <h5>■ Información General</h5>
95
96                                 <div class="mb-3">
97                                     <label class="form-label required">Nombre del Año Lectivo</label>
98                                     <input type="text"
99                                         name="name"
100                                         class="form-control"
101                                         placeholder="Ej: 2025-2026"
102                                         value="<?= htmlspecialchars($year['name']) ?> "
103                                         required>
104                                     <small class="text-muted">Formato recomendado: YYYY-YYYY</small>
105                                 </div>
106
107                                 <div class="row">
108                                     <div class="col-md-6 mb-3">
109                                         <label class="form-label required">Fecha de Inicio</label>
110                                         <input type="date"
111                                             name="start_date"
112                                             class="form-control"
113                                             value="<?= $year['start_date'] ?> "
114                                             required>
115                                     </div>
116                                     <div class="col-md-6 mb-3">
117                                         <label class="form-label required">Fecha de Fin</label>

```

```

118         <input type="date"
119             name="end_date"
120             class="form-control"
121             value="<?= $year['end_date'] ?>"
122             required>
123     </div>
124 </div>
125 </div>
126
127 <div class="alert alert-info">
128     <strong>■ Estado del año lectivo:</strong>
129     Para activar o desactivar este año lectivo, usa los botones
130     "Activar" o "Desactivar" en la tabla principal de años lectivos.
131 </div>
132
133 <div class="d-flex justify-content-between mt-4">
134     <a href="?action=academic" class="btn btn-secondary">
135         ← Cancelar
136     </a>
137     <button type="submit" class="btn btn-success">
138         ✓ Guardar Cambios
139     </button>
140 </div>
141 </form>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147
148 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
149 </body>
150 </html>

```


59. ArchImp/views/academic/enroll.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Matricular Estudiantes - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .container { max-width: 1200px; margin: 24px auto; padding: 0 16px; }
11         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot.
12         .form-group { margin-bottom: 20px; }
13         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
14         select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
15         .student-list { margin-top: 20px; }
16         .student-item { padding: 10px; border: 1px solid #ddd; border-radius: 4px; margin-bottom: 10px; display: flex; a.
17         .student-item input[type="checkbox"] { margin-right: 10px; }
18         button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointe.
19         button:hover { background: #218838; }
20         .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
21         .warning { background: #fff3cd; color: #856404; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
22         .info { background: #d1ecf1; color: #0c5460; padding: 10px; border-radius: 4px; margin-bottom: 15px; }
23         table { width: 100%; border-collapse: collapse; margin-top: 15px; }
24         th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; }
25         th { background: #f8f9fa; font-weight: bold; }
26         .badge { display: inline-block; padding: 4px 8px; border-radius: 3px; font-size: 12px; background: #007bff; colo.
27     </style>
28 </head>
29 <body>
30
31 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
32
33 <div class="breadcrumb">
34     <a href="?action=dashboard">Inicio</a> &rsaquo;
35     <a href="?action=academic">Configuración Académica</a> &rsaquo;
36     Matricular Estudiantes
37 </div>
38
39 <div class="container">
40
41     <div class="page-header" style="background: linear-gradient(135deg, #2e7d32, #388e3c);">
42         <div class="ph-icon"></div>
43     </div>
44     <h1>Matricular Estudiantes</h1>
45     <p>Asigna estudiantes a sus cursos del año lectivo activo</p>
46 </div>
47 </div>
48 <?php if(isset($_GET['enrolled'])): ?>
49     <div class="success">✓ <?=$_GET['enrolled'] ?> estudiante(s) matriculado(s) correctamente</div>
50 <?php endif; ?>
51
52 <?php if(isset($_GET['errors']) && $_GET['errors'] > 0): ?>
53     <div class="warning">■ <?=$_GET['errors'] ?> estudiante(s) no pudieron ser matriculados (ya est.
54 <?php endif; ?>
55
56 <?php if(isset($_GET['unenrolled'])): ?>
57     <div class="success">✓ Estudiante retirado del curso correctamente</div>
58 <?php endif; ?>
```

```

59
60 <?php if(isset($_GET['error']) && $_GET['error'] === 'not_enrolled'): ?>
61     <div class="warning">■ El estudiante no está matriculado en ningún curso</div>
62 <?php endif; ?>
63
64 <?php if(isset($_GET['error']) && $_GET['error'] === 'unenroll_failed'): ?>
65     <div class="warning">✗ Error al retirar el estudiante del curso</div>
66 <?php endif; ?>
67
68 <!-- Resumen de estudiantes -->
69 <div class="card">
70     <h2>Resumen</h2>
71     <div class="info">
72         <strong>Total estudiantes registrados:</strong> <?= count($allStudents) ?><br>
73         <strong>Ya matriculados:</strong> <?= count($allStudents) - count($availableStudents) ?>
74         <strong>Disponibles para matricular:</strong> <?= count($availableStudents) ?>
75     </div>
76 </div>
77
78 <!-- Formulario de matrícula -->
79 <?php if(count($availableStudents) > 0): ?>
80 <div class="card">
81     <h2>Matricular Estudiantes Disponibles</h2>
82     <p style="color: #666; margin-bottom: 20px;">
83         Solo se muestran estudiantes que aún NO están matriculados en ningún curso del año lectivo actual.
84     </p>
85
86     <form method="POST" action="?action=enroll_students">
87         <div class="form-group">
88             <label>Curso</label>
89             <select name="course_id" required>
90                 <option value="">Seleccionar curso...</option>
91                 <?php foreach($courses as $course): ?>
92                     <option value="<?= $course['id'] ?>">
93                         <?= $course['name'] ?> - <?= $course['shift_name'] ?>
94                     </option>
95                 <?php endforeach; ?>
96             </select>
97         </div>
98
99         <div class="student-list">
100             <label>Seleccionar Estudiantes</label>
101             <?php foreach($availableStudents as $student): ?>
102                 <div class="student-item">
103                     <input type="checkbox" name="student_ids[]" value="<?= $student['id'] ?>" id="student_&
104                     <label for="student_<?= $student['id'] ?>" style="margin: 0; font-weight: normal;">
105                         <?= $student['last_name'] . ' ' . $student['first_name'] ?>
106                         (<?= $student['dni'] ?? 'Sin cédula' ?>)
107                     </label>
108                 </div>
109             <?php endforeach; ?>
110         </div>
111
112         <button type="submit">Matricular Seleccionados</button>
113     </form>
114 </div>
115 <?php else: ?>
116 <div class="card">
117     <div class="warning">

```

```

118         ■ No hay estudiantes disponibles para matricular. Todos los estudiantes ya están asignados a cursos.
119     </div>
120 </div>
121 <?php endif; ?>
122
123 <!-- Lista de todos los estudiantes con su estado -->
124 <div class="card">
125     <h2>Estado de Todos los Estudiantes</h2>
126     <table>
127         <thead>
128             <tr>
129                 <th>#</th>
130                 <th>Apellidos y Nombres</th>
131                 <th>Cédula</th>
132                 <th>Curso</th>
133                 <th>Estado</th>
134                 <th>Acciones</th>
135             </tr>
136         </thead>
137         <tbody>
138             <?php
139             $counter = 1;
140             foreach($allStudents as $student):
141                 $course = $this->userModel->getStudentCourse($student['id'], $activeYear['id']);
142                 ?>
143                 <tr>
144                     <td><?php $counter++ ?></td>
145                     <td><?php $student['last_name'] . ' ' . $student['first_name'] ?></td>
146                     <td><?php $student['dni'] ?? '-' ?></td>
147                     <td>
148                         <?php if($course): ?>
149                             <strong><?php $course['name'] ?></strong>
150                             <br>
151                             <small style="color: #666;"><?php $course['shift_name'] ?></small>
152                         <?php else: ?>
153                             <span style="color: #999;">-</span>
154                         <?php endif; ?>
155                     </td>
156                     <td>
157                         <?php if($course): ?>
158                             <span class="badge" style="background: #28a745;">
159                                 ✓ Matriculado
160                             </span>
161                         <?php else: ?>
162                             <span class="badge" style="background: #6c757d;">Sin matricular</span>
163                         <?php endif; ?>
164                     </td>
165                     <td>
166                         <?php if($course): ?>
167                             <form method="POST" action="?action=unenroll_student" style="display: inline;"
168                                 onsubmit="return confirmUnenroll(event, '<?php addslashes($student['last_name']) ..
169                                 <input type="hidden" name="student_id" value="<?php $student['id'] ?>";
170                                 <button type="submit"
171                                     style="padding: 6px 12px; background: #dc3545; font-size: 13px;">
172                                 ✗ Retirar
173                             </button>
174                         </form>
175                         <?php else: ?>
176                             <span style="color: #999;">-</span>

```

```

177         <?php endif; ?>
178     </td>
179 </tr>
180 <?php endforeach; ?>
181 </tbody>
182 </table>
183 </div>
184 </div>
185
186 <script>
187 function confirmUnenroll(event, studentName, courseName) {
188     event.preventDefault();
189
190     const modal = document.createElement('div');
191     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
192
193     const modalContent = document.createElement('div');
194     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow: w
195
196     modalContent.innerHTML = `
197         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Retirar Estudiante del Curso</h3>
198         <p style="margin: 0 0 20px 0; color: #666;">
199             ¿Está seguro de retirar a <strong>${studentName}</strong> del curso <strong>${courseName}</strong>?
200         </p>
201         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #fff3cd; padding: 10px; border-ra
202             <strong>■ Advertencia:</strong> Esta acción quitará al estudiante del curso.
203             Los registros de asistencia se conservarán pero el estudiante no aparecerá como matriculado.
204         </p>
205         <div style="display: flex; gap: 10px; justify-content: flex-end;">
206             <button type="button" id="cancelUnenrollBtn" style="padding: 10px 20px; background: #6c757d; color: w
207                 Cancelar
208             </button>
209             <button type="button" id="confirmUnenrollBtn" style="padding: 10px 20px; background: #dc3545; color: .
210                 Sí, Retirar
211             </button>
212         </div>
213     `;
214
215     modal.appendChild(modalContent);
216     document.body.appendChild(modal);
217
218     const form = event.target;
219
220     document.getElementById('confirmUnenrollBtn').onclick = function() {
221         document.body.removeChild(modal);
222         form.submit();
223     };
224
225     document.getElementById('cancelUnenrollBtn').onclick = function() {
226         document.body.removeChild(modal);
227     };
228
229     return false;
230 }
231 </script>
232 </body>
233 </html>

```

60. ArchImp/views/academic/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Configuración Académica - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; }
11         .navbar h1 { font-size: 24px; }
12         .navbar a { color: white; text-decoration: none; margin-left: 20px; }
13         .container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
14         .grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(450px, 1fr)); gap: 20px; margin-bottom: 20px; }
15         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
16         .form-group { margin-bottom: 15px; }
17         .error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
18         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
19         input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
20         button { padding: 10px 20px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; }
21         button:hover { background: #218838; }
22         .btn-primary { background: #007bff; }
23         .btn-primary:hover { background: #0056b3; }
24         table { width: 100%; border-collapse: collapse; margin-top: 15px; }
25         th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
26         th { background: #f8f9fa; font-weight: bold; }
27         .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
28         h2 { margin-bottom: 20px; color: #333; }
29     </style>
30 </head>
31 <body>
32     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
33     <div class="breadcrumb">
34         <a href="?action=dashboard">Inicio & & Configuración Académica
35     </div>
36
37     <div class="container">
38
39         <div class="page-header" style="background: linear-gradient(135deg, #1a237e, #283593);">
40             <div class="ph-icon"></div>
41             <div>
42                 <h1>Configuración Académica</h1>
43                 <p>Años lectivos, cursos y asignaturas</p>
44             </div>
45         </div>
46
47         <?php if(isset($_GET['course_success'])): ?>
48             <div class="success">✓ Curso creado correctamente.
49                 <?php if(isset($_GET['subjects_loaded']) & (int)$_GET['subjects_loaded'] > 0): ?>
50                     Se pre-cargaron <strong><?php echo (int)$_GET['subjects_loaded']; ?> asignaturas</strong>.
51                 <?php elseif(isset($_GET['subjects_loaded'])): ?>
52                     Las asignaturas de este nivel ya estaban registradas.
53                 <?php endif; ?>
54             </div>
55         <?php endif; ?>
56
57         <?php if(isset($_GET['course_updated'])): ?>
58             <div class="success">✓ Curso actualizado correctamente</div>
59         <?php endif; ?>
```

```

59     <?php if(isset($_GET['course_deleted'])): ?>
60         <div class="success"><✓ Curso eliminado correctamente</div>
61     <?php endif; ?>
62
63     <?php if(isset($_GET['subject_success'])): ?>
64         <div class="success"><✓ Asignatura creada correctamente</div>
65     <?php endif; ?>
66     <?php if(isset($_GET['subject_updated'])): ?>
67         <div class="success"><✓ Asignatura actualizada correctamente</div>
68     <?php endif; ?>
69     <?php if(isset($_GET['subject_deleted'])): ?>
70         <div class="success"><✓ Asignatura eliminada correctamente</div>
71     <?php endif; ?>
72
73     <?php if(isset($_GET['sy_created'])): ?>
74         <div class="success"><✓ Año lectivo creado correctamente</div>
75     <?php endif; ?>
76     <?php if(isset($_GET['sy_updated'])): ?>
77         <div class="success"><✓ Año lectivo actualizado correctamente</div>
78     <?php endif; ?>
79     <?php if(isset($_GET['sy_deleted'])): ?>
80         <div class="success"><✓ Año lectivo eliminado correctamente</div>
81     <?php endif; ?>
82     <?php if(isset($_GET['sy_activated'])): ?>
83         <div class="success"><✓ Año lectivo activado correctamente</div>
84     <?php endif; ?>
85     <?php if(isset($_GET['sy_deactivated'])): ?>
86         <div class="success"><✓ Año lectivo desactivado correctamente</div>
87     <?php endif; ?>
88
89     <?php if(isset($_GET['error']) && $_GET['error'] === 'no_active_year'): ?>
90         <div class="error"><✗ No hay un año lectivo activo</div>
91     <?php endif; ?>
92     <?php if(isset($_GET['error']) && $_GET['error'] === 'course_not_found'): ?>
93         <div class="error"><✗ Curso no encontrado</div>
94     <?php endif; ?>
95     <?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_students'): ?>
96         <div class="error"><✗ No se puede eliminar el curso porque tiene estudiantes matriculados</div>
97     <?php endif; ?>
98     <?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_assignments'): ?>
99         <div class="error"><✗ No se puede eliminar el curso porque tiene asignaciones docentes</div>
100    <?php endif; ?>
101    <?php if(isset($_GET['error']) && $_GET['error'] === 'course_duplicate'): ?>
102        <div class="error"><✗ Ya existe un curso con ese nivel, paralelo y jornada en el año lectivo activo. El.
103    <?php endif; ?>
104    <?php if(isset($_GET['error']) && $_GET['error'] === 'subject_not_found'): ?>
105        <div class="error"><✗ Asignatura no encontrada</div>
106    <?php endif; ?>
107    <?php if(isset($_GET['error']) && $_GET['error'] === 'subject_has_assignments'): ?>
108        <div class="error"><✗ No se puede eliminar la asignatura porque tiene asignaciones docentes</div>
109    <?php endif; ?>
110    <?php if(isset($_GET['error']) && $_GET['error'] === 'cannot_delete_active'): ?>
111        <div class="error"><✗ No se puede eliminar el año lectivo activo</div>
112    <?php endif; ?>
113    <?php if(isset($_GET['error']) && $_GET['error'] === 'has_courses'): ?>
114        <div class="error"><✗ No se puede eliminar el año lectivo porque tiene cursos asociados</div>
115    <?php endif; ?>
116    <?php if(isset($_GET['error']) && $_GET['error'] === 'year_not_found'): ?>
117        <div class="error"><✗ Año lectivo no encontrado</div>

```

```

118         <?php endif; ?>
119
120     <!-- Años Lectivos -->
121     <div class="card" style="margin-bottom: 20px;">
122         <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px;">
123             <h2 style="margin: 0;">■ Años Lectivos</h2>
124             <button onclick="location.href='?action=create_school_year'" class="btn-primary" style="margin: 0;"&am
125                 + Crear Año Lectivo
126             </button>
127         </div>
128
129         <table>
130             <thead>
131                 <tr>
132                     <th>#</th>
133                     <th>Nombre</th>
134                     <th>Fecha Inicio</th>
135                     <th>Fecha Fin</th>
136                     <th>Estado</th>
137                     <th>Acciones</th>
138                 </tr>
139             </thead>
140             <tbody>
141                 <?php if(empty($schoolYears)): ?>
142                     <tr>
143                         <td colspan="6" style="text-align:center; color: #999;">No hay años lectivos registrad.
144                     </tr>
145                 <?php else: ?>
146                     <?php $counter = 1; foreach($schoolYears as $year): ?>
147                         <tr style="<?php $year['is_active'] ? 'background: #e7f3ff;' : '' ?>">
148                             <td><?php $counter++ ?></td>
149                             <td><strong><?php htmlspecialchars($year['name']) ?></strong></td>
150                             <td><?php date('d/m/Y', strtotime($year['start_date'])) ?></td>
151                             <td><?php date('d/m/Y', strtotime($year['end_date'])) ?></td>
152                             <td>
153                                 <?php if($year['is_active']): ?>
154                                     <span style="background: #28a745; color: white; padding: 4px 10px; border-radius: .
155                                     ✓ ACTIVO
156                                 </span>
157                                 <?php else: ?>
158                                     <span style="background: #6c757d; color: white; padding: 4px 10px; border-radius: .
159                                     Inactivo
160                                 </span>
161                                 <?php endif; ?>
162                             </td>
163                             <td style="white-space: nowrap;">
164                                 <button onclick="location.href='?action=edit_school_year&id=<?php $year['id'] ?&a
165                                     style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #212529;">
166                                     ─■ Editar
167                                 </button>
168
169                                 <?php if($year['is_active']): ?>
170                                     <form method="POST" action="?action=deactivate_school_year" style="display: inlin.
171                                     onsubmit="return confirmSY(event,'desactivar','<?php htmlspecialchars(addslas.
172                                     <input type="hidden" name="year_id" value="<?php $year['id'] ?>">
173                                     <button type="submit" style="padding: 5px 10px; font-size: 12px; background: .
174                                     ■ Desactivar
175                                 </button>
176                             </td>

```

```

177         <?php else: ?>
178             <form method="POST" action="?action=activate_school_year" style="display: inline;
179                 onsubmit="return confirmSY(event, 'activar', '<?= htmlspecialchars(addslashes.
180                 <input type="hidden" name="year_id" value="<? $year['id'] ?>";>
181                 <button type="submit" style="padding: 5px 10px; font-size: 12px; background: .
182                     ✓ Activar
183                 </button>
184             </form>
185
186             <form method="POST" action="?action=delete_school_year" style="display: inline;"
187                 onsubmit="return confirmDelete(event, '<?= htmlspecialchars($year['name']) .
188                 <input type="hidden" name="year_id" value="<? $year['id'] ?>";>
189                 <button type="submit" style="padding: 5px 10px; font-size: 12px; background: .
190                     ■■ Eliminar
191                 </button>
192             </form>
193         <?php endif: ?>
194     </td>
195 </tr>
196 <?php endforeach: ?>
197 <?php endif: ?>
198 </tbody>
199 </table>
200 </div>
201
202 <div class="grid">
203     <!-- Crear Curso -->
204     <div class="card">
205         <div style="display:flex; align-items:center; gap:10px; margin-bottom:20px; padding-bottom:16px; bord.
206         <div style="width:36px;height:36px;background:linear-gradient(135deg,#1a237e,#283593);border-radi.
207         </div>
208         <h2 style="margin:0;font-size:17px;color:#1a237e;">Crear Nuevo Curso</h2>
209         <p style="margin:0;font-size:12px;color:#888;">Las asignaturas se cargan automáticamente s.
210         </div>
211     </div>
212
213     <form method="POST" action="?action=create_course" id="courseForm">
214     <?php
215     $cf = $_SESSION['course_form'] ?? [];
216     unset($_SESSION['course_form']);
217     ?>
218
219     <!-- Fila 1: Nivel + Grado -->
220     <div style="display:grid;grid-template-columns:1fr 1fr;gap:14px;margin-bottom:14px;">
221         <div class="form-group" style="margin:0;">
222             <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
223             <select name="education_type" id="education_type" required onchange="updateGradeLevels()"
224                 style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-siz.
225                 onfocus="this.style.borderColor='#1a237e'" onblur="this.style.borderColor='#e0e0e0'">
226                 <option value="">Seleccionar...</option>
227                 <option value="inicial" <?= ($cf['education_type']??'')=='inicial'? 'selected':'' .
228                 <option value="egb" <?= ($cf['education_type']??'')=='egb' ? 'selected':'' .
229                 <option value="bgu" <?= ($cf['education_type']??'')=='bgu' ? 'selected':'' .
230                 <option value="bt" <?= ($cf['education_type']??'')=='bt' ? 'selected':'' .
231             </select>
232         </div>
233         <div class="form-group" id="group_grade" style="margin:0;">
234             <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
235             <select name="grade_level" id="grade_level" required onchange="onGradeChange()"

```



```

236         style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-siz.
237         onfocus="this.style.borderColor='#1a237e' " onblur="this.style.borderColor='#e0e0e0' ">
238         <option value=" ">Seleccione nivel primero...</option>
239         <?php if(!empty($cf['grade_level'])): ?>
240         <option value="<?>= htmlspecialchars($cf['grade_level']) ?>" selected>
241             <?>= htmlspecialchars($cf['grade_level']) ?>
242         </option>
243         <?php endif; ?>
244     </select>
245 </div>
246 </div>
247
248 <!-- BT: Figura + Carrera (ocultas por defecto) -->
249 <div id="group_specialty" style="display:none;margin-bottom:14px;">
250     <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
251     <select name="specialty" id="specialty" onchange="updateCarreras()"
252         style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-size:13.
253         <option value=" ">Seleccionar figura...</option>
254         <option value="Informática">Informática</option>
255         <option value="Administración">Administración</option>
256         <option value="Contabilidad">Contabilidad</option>
257         <option value="Comercialización y Ventas">Comercialización y Ventas</option>
258         <option value="Servicios Hoteleros">Servicios Hoteleros</option>
259         <option value="Turismo">Turismo</option>
260         <option value="Electromecánica Automotriz">Electromecánica Automotriz</option>
261         <option value="Instalaciones Eléctricas">Instalaciones Eléctricas</option>
262         <option value="Electrónica de Consumo">Electrónica de Consumo</option>
263         <option value="Mecanizado y Construcciones Metálicas">Mecanizado y Construcciones Metá.
264         <option value="Producción Agropecuaria">Producción Agropecuaria</option>
265         <option value="Acuicultura">Acuicultura</option>
266         <option value="Industrialización de Alimentos">Industrialización de Alimentos</opti.
267         <option value="Confección Textil">Confección Textil</option>
268         <option value="Música">Música</option>
269         <option value="Artes Plásticas">Artes Plásticas</option>
270         <option value="Diseño Gráfico">Diseño Gráfico</option>
271         <option value="Servicios de Belleza">Servicios de Belleza</option>
272         <option value="Atención Integral en Salud">Atención Integral en Salud</option>
273         <option value="Mecatrónica">Mecatrónica</option>
274         <option value="Refrigeración y Climatización">Refrigeración y Climatización</option.
275         <option value="Construcción">Construcción</option>
276         <option value="Redes y Telecomunicaciones">Redes y Telecomunicaciones</option>
277         <option value="Seguridad Industrial">Seguridad Industrial</option>
278         <option value="Logística y Transporte">Logística y Transporte</option>
279     </select>
280 </div>
281 <div id="group_carrera" style="display:none;margin-bottom:14px;">
282     <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
283     <select name="carrera" id="carrera" onchange="generateCourseName()"
284         style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-size:13.
285         <option value=" ">Sin especificar</option>
286     </select>
287 </div>
288
289 <!-- Fila 2: Paralelo + Jornada -->
290 <div style="display:grid;grid-template-columns:1fr 1fr;gap:14px;margin-bottom:14px;">
291     <div>
292         <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
293         <select name="parallel" id="parallel" required onchange="generateCourseName()"
294             style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-siz.

```

```

295         onfocus="this.style.borderColor='#1a237e' " onblur="this.style.borderColor='#e0e0e0' ">
296         </option value=" ">Seleccionar...</option>
297         <?php foreach(range('A', 'J') as $letter): ?>
298             <option value="<? $letter ?>" <? $ ( $cf['parallel']??'')==$letter?'sele.
299         <?php endforeach; ?>
300     </select>
301 </div>
302 </div>
303     <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
304     <select name="shift_id" id="shift_id" required onchange="generateCourseName()"
305         style="width:100%;padding:9px 12px;border:1.5px solid #e0e0e0;border-radius:7px;font-siz.
306         onfocus="this.style.borderColor='#1a237e' " onblur="this.style.borderColor='#e0e0e0' ">
307         <option value=" ">Seleccionar...</option>
308         <?php foreach($shifts as $shift): ?>
309             <option value="<? $shift['id'] ?>" data-shift="<? $shift['name'] ?>".
310                 <? $ ( $cf['shift_id']??'')==$shift['id']?'selected':'' ?>&
311                 <? ucfirst($shift['name']) ?>
312             </option>
313         <?php endforeach; ?>
314     </select>
315 </div>
316 </div>
317
318 <!-- Nombre generado -->
319 <div style="margin-bottom:18px;">
320     <label style="font-size:12px;font-weight:600;color:#555;margin-bottom:5px;display:block;">
321         Nombre del Curso
322     <span style="font-weight:400;color:#aaa;margin-left:4px;"><← generado automáticamente&a
323 </label>
324 <div style="position:relative;">
325     <input type="text" name="name" id="course_name" readonly
326         value="<? htmlspecialchars($cf['name'] ?? ' ') ?>"
327         style="width:100%;padding:9px 12px 9px 36px;border:1.5px solid #e0e0e0;border-radius:7px.
328     <span style="position:absolute;left:11px;top:50%;transform:translateY(-50%);font-size:15px.
329 </div>
330 </div>
331
332 <button type="submit"
333     style="width:100%;padding:11px;background:linear-gradient(135deg,#1a237e,#283593);color:white;bo.
334     onmouseover="this.style.opacity='.88' " onmouseout="this.style.opacity='1' ">
335     ■ Crear Curso
336 </button>
337 </form>
338 </div>
339
340 </script>
341 const gradeLevels = {
342     inicial: [
343         'Inicial 1 (0-3 años)',
344         'Inicial 2 (3-5 años)'
345     ],
346     egb: [
347         '1.º EGB - Preparatoria',
348         '2.º EGB', '3.º EGB', '4.º EGB',
349         '5.º EGB', '6.º EGB', '7.º EGB',
350         '8.º EGB', '9.º EGB', '10.º EGB'
351     ],
352     bgu: ['1.º BGU', '2.º BGU', '3.º BGU'],
353     bt: ['1.º BT', '2.º BT', '3.º BT']

```

```

354     };
355
356     // Grados EGB donde se permite jornada nocturna
357     const egbNocturnaAllowed = ['8.º EGB', '9.º EGB', '10.º EGB'];
358
359     const carreras = {
360         'Informática': ['Aplicaciones Informáticas', 'Programación de Software', 'Soporte Técnico', 'Sistemas Micro.'],
361         'Administración': ['Asistencia Administrativa', 'Gestión Empresarial'],
362         'Contabilidad': ['Contabilidad', 'Ventas e Información Comercial'],
363         'Comercialización y Ventas': ['Ventas e Información Comercial', 'Marketing'],
364         'Servicios Hoteleros': ['Hotelería', 'Hospitalidad'],
365         'Turismo': ['Turismo', 'Guía Turístico'],
366         'Electromecánica Automotriz': ['Electromecánica Automotriz', 'Mecánica Automotriz'],
367         'Instalaciones Eléctricas': ['Electricidad', 'Instalaciones Eléctricas'],
368         'Electrónica de Consumo': ['Electrónica', 'Mantenimiento Electrónico'],
369         'Atención Integral en Salud': ['Atención en Enfermería', 'Auxiliar de Salud'],
370         'Producción Agropecuaria': ['Producción Agropecuaria', 'Agroindustria'],
371         'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
372         'Diseño Gráfico': ['Diseño Gráfico', 'Multimedia'],
373         'Servicios de Belleza': ['Peluquería', 'Cosmetología']
374     };
375
376     function getNocturnaOption() {
377         const opts = document.getElementById('shift_id').options;
378         for (let i = 0; i < opts.length; i++) {
379             if ((opts[i].getAttribute('data-shift') || '').toLowerCase() === 'nocturna') return opts[i];
380         }
381         return null;
382     }
383
384     function updateNocturnaVisibility(type, grade) {
385         const nocturna = getNocturnaOption();
386         if (!nocturna) return;
387
388         let visible = false;
389         if (type === 'bgu' || type === 'bt') {
390             visible = true;
391         } else if (type === 'egb' && egbNocturnaAllowed.includes(grade)) {
392             visible = true;
393         }
394
395         nocturna.style.display = visible ? '' : 'none';
396
397         // Si estaba seleccionada y ya no aplica, limpiar
398         if (!visible && nocturna.selected) {
399             document.getElementById('shift_id').value = '';
400             generateCourseName();
401         }
402     }
403
404     function updateGradeLevels() {
405         const type = document.getElementById('education_type').value;
406         const gradeSelect = document.getElementById('grade_level');
407
408         gradeSelect.innerHTML = '<option value="">Seleccionar grado...</option>';
409         document.getElementById('course_name').value = '';
410
411         if (!type) return;
412     }

```

```

413         (gradeLevels[type] || []).forEach(function(g) {
414             const opt = document.createElement('option');
415             opt.value = g;
416             opt.textContent = g;
417             gradeSelect.appendChild(opt);
418         });
419
420         // Mostrar/ocultar campos BT
421         const isBT = (type === 'bt');
422         document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
423         document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
424         document.getElementById('specialty').required = isBT;
425
426         if (!isBT) {
427             document.getElementById('specialty').value = '';
428             document.getElementById('carrera').innerHTML = '<option value="">Sin especificar</option>';
429         }
430
431         updateNocturnaVisibility(type, '');
432         generateCourseName();
433     }
434
435     function onGradeChange() {
436         const type = document.getElementById('education_type').value;
437         const grade = document.getElementById('grade_level').value;
438         updateNocturnaVisibility(type, grade);
439         generateCourseName();
440     }
441
442     function updateCarreras() {
443         const fig = document.getElementById('specialty').value;
444         const carreraSelect = document.getElementById('carrera');
445         carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
446
447         if (carreras[fig]) {
448             carreras[fig].forEach(function(c) {
449                 const opt = document.createElement('option');
450                 opt.value = c;
451                 opt.textContent = c;
452                 carreraSelect.appendChild(opt);
453             });
454         }
455         generateCourseName();
456     }
457
458     function generateCourseName() {
459         const grade = document.getElementById('grade_level').value;
460         const parallel = document.getElementById('parallel').value;
461         const shiftSelect = document.getElementById('shift_id');
462         const shiftOpt = shiftSelect.options[shiftSelect.selectedIndex];
463         const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';
464         const type = document.getElementById('education_type').value;
465         const specialty = document.getElementById('specialty').value;
466         const carrera = document.getElementById('carrera').value;
467
468         if (!grade || !parallel || !shiftName) {
469             document.getElementById('course_name').value = '';
470             return;
471         }

```

[illegible]

```

531         &lt;td&gt;&lt;?=$course['id'] ?&gt;&lt;/td&gt;
532         &lt;td&gt;&lt;?=$course['name'] ?&gt;&lt;/td&gt;
533         &lt;td&gt;&lt;?=$course['grade_level'] ?&gt;&lt;/td&gt;
534         &lt;td&gt;&lt;?=$course['parallel'] ?&gt;&lt;/td&gt;
535         &lt;td&gt;&lt;?=$course['shift_name'] ?&gt;&lt;/td&gt;
536         &lt;td style="white-space: nowrap;"&gt;
537             &lt;button onclick="location.href='?action=course_subjects&course_id=&lt;?=$course['id']'
538                 style="padding: 5px 10px; font-size: 12px; background: #6f42c1; color:white;"&gt;
539                 ■ Asignaturas
540             &lt;/button&gt;
541             &lt;button onclick="location.href='?action=view_course_students&course_id=&lt;?=$course['id']'
542                 style="padding: 5px 10px; font-size: 12px; background: #007bff;"&gt;
543                 ■ Estudiantes
544             &lt;/button&gt;
545             &lt;button onclick="location.href='?action=edit_course&id=&lt;?=$course['id'] ?&gt;"
546                 style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #000;"&gt;
547                 =■ Editar
548             &lt;/button&gt;
549             &lt;form method="POST" action="?action=delete_course" style="display: inline;"
550                 onsubmit="return confirmDeleteCourse(event, '&lt;?=$course['id'] ?&gt;'"&gt;
551                 &lt;input type="hidden" name="course_id" value="&lt;?=$course['id'] ?&gt;"&gt;
552                 &lt;button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;"&gt;
553                     ■■ Eliminar
554                 &lt;/button&gt;
555             &lt;/form&gt;
556         &lt;/td&gt;
557     &lt;/tr&gt;
558 &lt;?php endforeach; ?&gt;
559 &lt;/tbody&gt;
560 &lt;/table&gt;
561     &lt;button onclick="location.href='?action=enroll_students'" class="btn-primary" style="margin-top: 15px;"&gt;
562         Matricular Estudiantes
563     &lt;/button&gt;
564 &lt;/div&gt;
565
566 &lt;/div&gt;
567
568 &lt;script&gt;
569 function confirmDeleteCourse(event, courseName) {
570     event.preventDefault();
571     const modal = document.createElement('div');
572     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
573     const modalContent = document.createElement('div');
574     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow: 0 0 10px #000;';
575     modalContent.innerHTML = `
576         &lt;h3 style="margin: 0 0 15px 0; color: #dc3545;"&gt;■■ Eliminar Curso&lt;/h3&gt;
577         &lt;p style="margin: 0 0 20px 0; color: #666;"&gt;¿Está seguro de eliminar el curso &lt;strong&gt;${courseName}&lt;/strong&gt;?&lt;/p&gt;
578         &lt;p style="margin: 0 0 20px 0; color: #666; font-size: 14px;"&gt;&lt;strong&gt;Nota:&lt;/strong&gt; No se puede eliminar un curso que ya tiene estudiantes.&lt;/p&gt;
579         &lt;div style="display: flex; gap: 10px; justify-content: flex-end;"&gt;
580             &lt;button type="button" id="cancelCourseBtn" style="padding: 10px 20px; background: #6c757d; color: white;"&gt;Cancelar
581             &lt;button type="button" id="confirmCourseBtn" style="padding: 10px 20px; background: #dc3545; color: white;"&gt;Eliminar
582         &lt;/div&gt;`;
583     modal.appendChild(modalContent);
584     document.body.appendChild(modal);
585     const form = event.target;
586     document.getElementById('confirmCourseBtn').onclick = function() { document.body.removeChild(modal); form.submit(); };
587     document.getElementById('cancelCourseBtn').onclick = function() { document.body.removeChild(modal); };
588     return false;
589 }

```

```

590
591
592     function confirmDelete(event, yearName) {
593         event.preventDefault();
594         const modal = document.createElement('div');
595         modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
596         const modalContent = document.createElement('div');
597         modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow: 0 0 10px #000;';
598         modalContent.innerHTML = `
599             <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Año Lectivo</h3>
600             <p style="margin: 0 0 20px 0; color: #666;">¿Está seguro de eliminar el año lectivo <strong>${yearName}</strong>?</p>
601             <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;"><strong>Nota:</strong> No se puede eliminar el año lectivo actual.</p>
602             <div style="display: flex; gap: 10px; justify-content: flex-end;">
603                 <button type="button" id="cancelYearBtn" style="padding: 10px 20px; background: #6c757d; color: white;">Cancelar</button>
604                 <button type="button" id="confirmYearBtn" style="padding: 10px 20px; background: #dc3545; color: white;">Eliminar</button>
605             </div>`;
606         modal.appendChild(modalContent);
607         document.body.appendChild(modal);
608         const form = event.target;
609         document.getElementById('confirmYearBtn').onclick = function() { document.body.removeChild(modal); form.submit(); };
610         document.getElementById('cancelYearBtn').onclick = function() { document.body.removeChild(modal); };
611         return false;
612     }
613
614     // Modal para activar/desactivar año lectivo
615     function confirmSY(event, accion, nombre) {
616         event.preventDefault();
617         var form = event.target;
618         var esActivar = accion === 'activar';
619         var modal = document.createElement('div');
620         modal.style.cssText = 'position:fixed;top:0;left:0;width:100%;height:100%;background:rgba(0,0,0,.5);display:flex;align-items:center;justify-content:center;';
621         var color = esActivar ? '#28a745' : '#6c757d';
622         var icono = esActivar ? '✓' : '■';
623         var btnTxt = esActivar ? 'Sí, Activar' : 'Sí, Desactivar';
624         var desc = esActivar
625             ? 'Se desactivará el año lectivo actual y se activará <strong>${nombre}</strong>' + nombre + '</strong>.'
626             : '¿Seguro que deseas desactivar <strong>${nombre}</strong>' + nombre + '</strong>?';
627         modal.innerHTML = `<div style="background:white;padding:30px;border-radius:8px;max-width:460px;width:90%;box-shadow:0 0 10px #000;">
628             <h3 style="margin:0 0 12px;color:' + color + ';"><strong>${desc}</strong></h3>
629             <p style="color:#555;margin-bottom:20px;"><strong>${desc}</strong></p>
630             <div style="display:flex;gap:10px;justify-content:flex-end;">
631                 <button id="btnCancelSY" style="padding:9px 18px;background:#6c757d;color:white;border:none;border-radius:4px;">Cancelar</button>
632                 <button id="btnConfirmSY" style="padding:9px 18px;background:' + color + ';color:white;border:none;border-radius:4px;">${btnTxt}</button>
633             </div>
634         </div>`;
635         document.body.appendChild(modal);
636         document.getElementById('btnConfirmSY').onclick = function(){ document.body.removeChild(modal); form.submit(); };
637         document.getElementById('btnCancelSY').onclick = function(){ document.body.removeChild(modal); };
638         return false;
639     }
640 </script>
641 </body>
642 </html>

```

61. ArchImp/views/academic/school_year_create.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Crear Año Lectivo - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      <a href="?action=academic">Configuración Académica</a> &rsquo;
16      Crear Año Lectivo
17  </div>
18
19  <div class="container" style="max-width:600px;">
20
21      <div class="page-header" style="background:linear-gradient(135deg,#1a237e,#283593);">
22          <div class="ph-icon">■</div>
23          <div>
24              <h1>Crear Año Lectivo</h1>
25              <p>Define el período académico</p>
26          </div>
27      </div>
28
29      <?php if(!empty($errors)): ?>
30          <div class="alert alert-danger">
31              <?php foreach($errors as $e): ?>
32                  <div>■ <?= htmlspecialchars($e) ?></div>
33              <?php endforeach; ?>
34          </div>
35      <?php endif; ?>
36
37      <div class="panel">
38          <form method="POST" action="?action=create_school_year">
39
40              <div class="form-group">
41                  <label>Nombre del Año Lectivo *</label>
42                  <input type="text" name="name" class="form-control" required
43                      value="<?= htmlspecialchars($_POST['name']) ?? '' ?>"
44                      placeholder="Ej: 2025-2026">
45                  <small style="color:#888;">Formato recomendado: AAAA-AAAA</small>
46              </div>
47
48              <div class="form-row">
49                  <div class="form-group">
50                      <label>Fecha de Inicio *</label>
51                      <input type="date" name="start_date" class="form-control" required
52                          value="<?= htmlspecialchars($_POST['start_date']) ?? '' ?>">
53                  </div>
54                  <div class="form-group">
55                      <label>Fecha de Fin *</label>
56                      <input type="date" name="end_date" class="form-control" required
57                          value="<?= htmlspecialchars($_POST['end_date']) ?? '' ?>">
58                  </div>
59              </div>
60          </form>
61      </div>
62  </div>
63  </body>
64  </html>
```



```

59         </div>
60
61     <div class="form-group" style="margin-top:8px;">
62         <label style="display:flex;align-items:center;gap:10px;cursor:pointer;font-weight:normal;">
63             <input type="checkbox" name="is_active" value="1"
64                 <?= isset($_POST['is_active']) ? 'checked' : '' ?>
65                 style="width:18px;height:18px;accent-color:#28a745;">
66             <span>
67                 <strong>Marcar como año lectivo activo</strong><br>
68                 <small style="color:#888;">Solo puede haber un año lectivo activo a la vez</small>
69             </span>
70         </label>
71     </div>
72
73     <div style="display:flex;gap:10px;margin-top:20px;">
74         <button type="submit" class="btn btn-success">■ Crear Año Lectivo</button>
75         <a href="?action=academic" class="btn btn-outline">Cancelar</a>
76     </div>
77
78     </form>
79 </div>
80
81 </div>
82 </body>
83 </html>

```

62. ArchImp/views/academic/school_year_edit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Editar Año Lectivo - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">Inicio</a> &rsquo;
15      <a href="?action=academic">Configuración Académica</a> &rsquo;
16      Editar Año Lectivo
17  </div>
18
19  <div class="container" style="max-width:600px;">
20
21      <div class="page-header" style="background:linear-gradient(135deg,#e65100,#ef6c00);">
22          <div class="ph-icon">■</div>
23          <div>
24              <h1>Editar Año Lectivo</h1>
25              <p><?= htmlspecialchars($year['name']) ?></p>
26          </div>
27      </div>
28
29      <?php if(!empty($errors)): ?>
30          <div class="alert alert-danger">
31              <?php foreach($errors as $e): ?>
32                  <div>■ <?= htmlspecialchars($e) ?></div>
33              <?php endforeach; ?>
34          </div>
35      <?php endif; ?>
36
37      <div class="panel">
38          <form method="POST" action="?action=edit_school_year&id=<?= $year['id'] ?>">
39
40              <div class="form-group">
41                  <label>Nombre del Año Lectivo *</label>
42                  <input type="text" name="name" class="form-control" required
43                      value="<?= htmlspecialchars($_POST['name']) ?? $year['name'] ?>"
44                      placeholder="Ej: 2025-2026">
45              </div>
46
47              <div class="form-row">
48                  <div class="form-group">
49                      <label>Fecha de Inicio *</label>
50                      <input type="date" name="start_date" class="form-control" required
51                          value="<?= htmlspecialchars($_POST['start_date']) ?? $year['start_date'] ?>">
52                  </div>
53                  <div class="form-group">
54                      <label>Fecha de Fin *</label>
55                      <input type="date" name="end_date" class="form-control" required
56                          value="<?= htmlspecialchars($_POST['end_date']) ?? $year['end_date'] ?>">
57                  </div>
58              </div>
59          </form>
60      </div>
61  </div>
62  </body>
63  </html>
```

```
59
60     <?php if($year['is_active']): ?>
61         <div class="alert alert-info" style="margin-top:8px;font-size:13px;">
62             ■ Este es el <strong>año lectivo activo</strong> actualmente. Para cambiarlo, activa otr.
63         </div>
64     <?php endif; ?>
65
66     <div style="display:flex;gap:10px;margin-top:20px;">
67         <button type="submit" class="btn btn-success">■ Guardar Cambios</button>
68         <a href="?action=academic" class="btn btn-outline">Cancelar</a>
69     </div>
70
71     </form>
72 </div>
73
74 </div>
75 </body>
76 </html>
```

63. ArchImp/views/academic/subject_edit.php

```
1  <?php
2  require_once BASE_PATH . '/helpers/Security.php';
3  Security::requireLogin();
4  if (!Security::hasRole('autoridad')) {
5      die('Acceso denegado');
6  }
7  ?>
8  <!DOCTYPE html>
9  <html lang="es">
10 <head>
11     <meta charset="UTF-8">
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <title>Editar Asignatura - EcuAsist</title>
14     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
15     <style>
16         body { padding-top: 70px; }
17     </style>
18 </head>
19 <body>
20     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
21
22     <div class="container">
23         <div class="row justify-content-center">
24             <div class="col-md-8">
25                 <div class="card shadow">
26                     <div class="card-header bg-primary text-white">
27                         <h4 class="mb-0">■ Editar Asignatura: <?= htmlspecialchars($subject['name']) ?><
28                     </div>
29                     <div class="card-body">
30                         <?php if (!empty($errors)): ?>
31                             <div class="alert alert-danger">
32                                 <?php foreach ($errors as $error): ?>
33                                     <div><?= htmlspecialchars($error) ?></div>
34                                 <?php endforeach; ?>
35                             </div>
36                         <?php endif; ?>
37
38                         <form method="POST">
39                             <div class="mb-3">
40                                 <label class="form-label">Nombre *</label>
41                                 <input type="text"
42                                     name="name"
43                                     class="form-control"
44                                     placeholder="Ej: Matemáticas"
45                                     value="<?= htmlspecialchars($subject['name']) ?>"
46                                     required>
47                             </div>
48
49                             <div class="mb-3">
50                                 <label class="form-label">Código *</label>
51                                 <input type="text"
52                                     name="code"
53                                     class="form-control"
54                                     placeholder="Ej: MAT"
55                                     value="<?= htmlspecialchars($subject['code']) ?>"
56                                     required>
57                                 <small class="text-muted">Código corto para identificar la asignatura</small>
58                             </div>
```

```
59
60         <div class="d-flex justify-content-between mt-4">
61             <a href="?action=academic" class="btn btn-secondary">← Cancelar</a>
62             <button type="submit" class="btn btn-success">✓ Guardar Cambios</button>
63         </div>
64     </form>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70
71     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
72 </body>
73 </html>
```

64. ArchImp/views/assignments/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Asignaciones Docentes - EcuAsist</title>
8      <style>
9          * { margin: 0; padding: 0; box-sizing: border-box; }
10         body { font-family: Arial, sans-serif; background: #f4f4f4; }
11         .container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot.
13         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; font-size: 13px; }
14         select { width: 100%; padding: 9px 10px; border: 1px solid #ddd; border-radius: 4px; font-size: 14px; }
15         table { width: 100%; border-collapse: collapse; margin-top: 15px; }
16         th, td { padding: 11px 13px; text-align: left; border-bottom: 1px solid #eee; font-size: 14px; }
17         th { background: #f8f9fa; font-weight: bold; color: #555; }
18         tr:hover td { background: #f9f9f9; }
19         .btn-danger { background: #dc3545; color: white; padding: 5px 12px; font-size: 12px; border: none; border-radius.
20         .btn-danger:hover { background: #c82333; }
21         .btn-clear { padding: 9px 16px; background: #6c757d; color: white; text-decoration: none; border-radius: 4px; fo.
22         .info-banner { background: #e8f4fd; border-left: 4px solid #007bff; padding: 14px 18px; border-radius: 4px; marg.
23         .alert-success { background: #d4edda; color: #155724; padding: 12px 16px; border-radius: 4px; margin-bottom: 16p.
24         .alert-error { background: #f8d7da; color: #721c24; padding: 12px 16px; border-radius: 4px; margin-bottom: 16p.
25         .empty-msg { text-align: center; color: #999; padding: 30px; }
26         .filter-grid { display: grid; grid-template-columns: repeat(3,1fr) auto; gap: 14px; align-items: flex-end; }
27         @media(max-width:700px) { .filter-grid { grid-template-columns: 1fr 1fr; } }
28     </style>
29 </head>
30 <body>
31
32 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
33
34 <div class="breadcrumb">
35     <a href="?action=dashboard">Inicio &rsquo; Asignaciones Docentes
36 </div>
37
38 <div class="container">
39
40     <div class="page-header" style="background:linear-gradient(135deg,#e65100,#f57c00);">
41         <div class="ph-icon">■</div>
42     </div>
43     <h1>Asignaciones Docente – Materia</h1>
44     <p>Consulta global de todas las asignaciones del año lectivo activo</p>
45 </div>
46 <div class="ph-actions">
47     <a href="?action=academic" style="padding:8px 16px;background:rgba(255,255,255,0.2);color:white;text-deco.
48         ■ Ir a Cursos
49     </a>
50     <a href="?action=tutor_management" style="padding:8px 16px;background:rgba(255,255,255,0.2);color:white;t.
51         ■■ Tutores
52     </a>
53 </div>
54 </div>
55
56 <?php if(isset($_GET['removed'])): ?>
57     <div class="alert-success">✓ Asignación eliminada correctamente.</div>
58 <?php endif; ?>
```

```

59     <?php if(isset($_GET['error'])): ?>
60         <div class="alert-error">X <?!= htmlspecialchars($_GET['error']) ?></div>
61     <?php endif; ?>
62
63     <div class="info-banner">
64         ■ Para <strong>asignar</strong> un docente a una materia, ve a
65         <a href="?action=academic" style="color:#0056b3;font-weight:bold;">Configuración Académica</a>
66         → selecciona un curso → botón <strong>■ Asignaturas</strong> → botón <strong>■ Asignar</strong>
67     </div>
68
69     <div class="card">
70         <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:18px;">
71             <h2 style="color:#333;font-size:18px;">■ Todas las asignaciones</h2>
72             <span style="font-size:13px;color:#888;"><?!= count($assignments) ?> registradas</span>
73         </div>
74
75         <form method="GET" action="">
76             <input type="hidden" name="action" value="assignments">
77             <div class="filter-grid">
78                 <div>
79                     <label>Filtrar por Curso</label>
80                     <select name="filter_course" onchange="this.form.submit()">
81                         <option value="">Todos los cursos</option>
82                         <?php foreach($courses as $c): ?>
83                             <option value="<?!= $c['id'] ?>" <?!= (!empty($_GET['filter_course']) && $_GET['filter_course'] != $c['id']) ?>
84                                 <?!= htmlspecialchars($c['name']) ?>
85                             </option>
86                         <?php endforeach; ?>
87                     </select>
88                 </div>
89                 <div>
90                     <label>Filtrar por Asignatura</label>
91                     <select name="filter_subject" onchange="this.form.submit()">
92                         <option value="">Todas las asignaturas</option>
93                         <?php foreach($subjects as $s): ?>
94                             <option value="<?!= $s['id'] ?>" <?!= (!empty($_GET['filter_subject']) && $_GET['filter_subject'] != $s['id']) ?>
95                                 <?!= htmlspecialchars($s['name']) ?>
96                             </option>
97                         <?php endforeach; ?>
98                     </select>
99                 </div>
100                 <div>
101                     <label>Filtrar por Docente</label>
102                     <select name="filter_teacher" onchange="this.form.submit()">
103                         <option value="">Todos los docentes</option>
104                         <?php foreach($teachers as $t): ?>
105                             <option value="<?!= $t['id'] ?>" <?!= (!empty($_GET['filter_teacher']) && $_GET['filter_teacher'] != $t['id']) ?>
106                                 <?!= htmlspecialchars($t['last_name'].' '.$t['first_name']) ?>
107                             </option>
108                         <?php endforeach; ?>
109                     </select>
110                 </div>
111                 <?php if(!empty($_GET['filter_course']) || !empty($_GET['filter_subject']) || !empty($_GET['filter_teacher'])):
112                     <div style="padding-top:18px;">
113                         <a href="?action=assignments" class="btn-clear">X Limpiar</a>
114                     </div>
115                 <?php endif; ?>
116             </div>
117         </form>

```

```

118
119     <table>
120         <thead>
121             <tr>
122                 <th>Curso</th>
123                 <th>Asignatura</th>
124                 <th>Docente</th>
125                 <th style="width:110px;text-align:center;">Acción</th>
126             </tr>
127         </thead>
128         <tbody>
129             <?php
130                 $list = $assignments;
131                 if (!empty($_GET['filter_course'])) {
132                     $id = (int)$_GET['filter_course']; $nm = '';
133                     foreach($courses as $c){ if($c['id']==$id){ $nm=$c['name']; break; } }
134                     $list = array_filter($list, fn($a) => $a['course_name']==$nm);
135                 }
136                 if (!empty($_GET['filter_subject'])) {
137                     $id = (int)$_GET['filter_subject']; $nm = '';
138                     foreach($subjects as $s){ if($s['id']==$id){ $nm=$s['name']; break; } }
139                     $list = array_filter($list, fn($a) => $a['subject_name']==$nm);
140                 }
141                 if (!empty($_GET['filter_teacher'])) {
142                     $id = (int)$_GET['filter_teacher']; $nm = '';
143                     foreach($teachers as $t){ if($t['id']==$id){ $nm=$t['last_name'].' '.$t['first_name']; break; } }
144                     $list = array_filter($list, fn($a) => $a['teacher_name']==$nm);
145                 }
146             ?>
147             <?php if(empty($list)): ?>
148                 <tr><td colspan="4" class="empty-msg">No hay asignaciones que coincidan con los filtros..
149             <?php else: ?>
150                 <?php foreach($list as $a): ?>
151                     <tr>
152                         <td><?= htmlspecialchars($a['course_name']) ?></td>
153                         <td><?= htmlspecialchars($a['subject_name']) ?></td>
154                         <td><?= htmlspecialchars($a['teacher_name']) ?></td>
155                         <td style="text-align:center;">
156                             <form method="POST" action="?action=remove_assignment" style="display:inline;">
157                                 <input type="hidden" name="assignment_id" value="<?= $a['id'] ?>">
158                                 <button type="submit" class="btn-danger"
159                                     onclick="return confirmDelete(event,'<?= htmlspecialchars(addslashes($a['teacher_
160                                     ■■ Quitar
161                                 </button>
162                             </form>
163                         </td>
164                     </tr>
165                 <?php endforeach; ?>
166             <?php endif; ?>
167         </tbody>
168     </table>
169 </div>
170 </div>
171
172 <script>
173 function confirmDelete(event, teacher, course, subject) {
174     event.preventDefault();
175     const m = document.createElement('div');
176     m.style.cssText = 'position:fixed;top:0;left:0;width:100%;height:100%;background:rgba(0,0,0,.5);display:flex;align-i.

```



```

177     m.innerHTML = `<div style="background:white;padding:28px;border-radius:8px;max-width:460px;width:90%;box-shadow:0.
178         <h3 style="color:#dc3545;margin-bottom:14px;">&#9633; Quitar asignación</h3>
179         <p style="color:#555;margin-bottom:6px;">&#x2602;<strong>Docente:</strong> ${teacher}</p>
180         <p style="color:#555;margin-bottom:6px;">&#x2602;<strong>Curso:</strong> ${course}</p>
181         <p style="color:#555;margin-bottom:18px;">&#x2602;<strong>Asignatura:</strong> ${subject}</p>
182         <div style="display:flex;gap:10px;justify-content:flex-end;">
183             <button id="bC" style="padding:8px 18px;background:#6c757d;color:white;border:none;border-radius:4px;cursor:
184             <button id="bO" style="padding:8px 18px;background:#dc3545;color:white;border:none;border-radius:4px;cursor:
185         </div>&#x2602;</div>`;
186     document.body.appendChild(m);
187     const form = event.target.closest('form');
188     document.getElementById('bO').onclick = () => { document.body.removeChild(m); form.submit(); };
189     document.getElementById('bC').onclick = () => document.body.removeChild(m);
190     return false;
191 }
192 </script>
193 </body>
194 </html>

```

65. ArchImp/views/assignments/tutor.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Asignar Docente Tutor - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
11         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot.
12         .form-group { margin-bottom: 15px; }
13         .error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
14         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
15         select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
16         button { padding: 10px 20px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointe.
17         button:hover { background: #0056b3; }
18         .btn-danger { background: #dc3545; }
19         .btn-danger:hover { background: #c82333; }
20         .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
21         table { width: 100%; border-collapse: collapse; margin-top: 15px; }
22         th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
23         th { background: #f8f9fa; font-weight: bold; }
24         h2 { margin-bottom: 20px; color: #333; }
25     </style>
26 </head>
27 <body>
28     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
29     <div class="breadcrumb">
30         <a href="?action=dashboard">Inicio &rsaquo; Asignar Docente Tutor
31     </div>
32
33     <div class="container">
34
35         <div class="page-header" style="background: linear-gradient(135deg, #6ab9a, #8e24aa);">
36             <div class="ph-icon">■■■■</div>
37             <div>
38                 <h1>Asignar Docente Tutor</h1>
39                 <p>Designa el tutor responsable de cada curso</p>
40             </div>
41         </div>
42
43         <?php if(isset($_GET['tutor_success'])): ?>
44             <div class="success">✓ Tutor asignado correctamente</div>
45         <?php endif; ?>
46         <?php if(isset($_GET['tutor_error'])): ?>
47             <div class="error">✗ <?= htmlspecialchars($_GET['tutor_error']) ?></div>
48         <?php endif; ?>
49         <?php if(isset($_GET['tutor_removed'])): ?>
50             <div class="success">✓ Tutor eliminado correctamente</div>
51         <?php endif; ?>
52
53         <div class="card">
54             <div class="header-actions" style="display: flex; justify-content: space-between; align-items: center; marg.
55                 <h2 style="margin: 0;">■■■■ Asignar Docente Tutor</h2>
56                 <a href="?action=assignments" style="padding: 8px 16px; background: #28a745; color: white; text-decorat.
57                 ■ Ir a Asignar Docente-Materia
58                 </a>
```

```

59         </div>
60         <form method="POST" action="?action=set_tutor" id="tutorForm">
61             <div class="form-group">
62                 <label>Curso</label>
63                 <select name="course_id" id="course_tutor" required>
64                     <option value="">Seleccionar...</option>
65                     <?php foreach($courses as $course): ?>
66                         <option value="<?php echo $course['id']; ?>">
67                             <?php echo $course['name']; ?> - <?php echo $course['shift_name']; ?>
68                         </option>
69                     <?php endforeach; ?>
70                 </select>
71             </div>
72
73             <div class="form-group">
74                 <label>Docente Tutor</label>
75                 <select name="teacher_id" id="teacher_tutor" required>
76                     <option value="">Primero seleccione un curso...</option>
77                 </select>
78             </div>
79
80             <button type="submit">Asignar Tutor</button>
81         </form>
82
83         <hr style="margin: 30px 0; border: none; border-top: 1px solid #ddd;">
84
85     </div>
86
87     <div class="card">
88         <h2>Gestión de Tutores por Curso</h2>
89         <table>
90             <thead>
91                 <tr>
92                     <th>Curso</th>
93                     <th>Docente Tutor</th>
94                     <th>Acción</th>
95                 </tr>
96             </thead>
97             <tbody>
98                 <?php foreach($courses as $course): ?>
99                     <?php
100                         // Buscar si el curso tiene tutor
101                         $courseTutor = null;
102                         foreach($assignments as $assignment) {
103                             if ($assignment['course_name'] == $course['name'] & $assignment['is_tutor']) {
104                                 $courseTutor = $assignment;
105                                 break;
106                             }
107                         }
108                     <?php
109                     <td><?php echo $course['name']; ?></td>
110                     <td>
111                         <?php if($courseTutor): ?>
112                             <?php echo $courseTutor['teacher_name']; ?>
113                         <?php else: ?>
114                             <em style="color: #999;">Sin tutor asignado</em>
115                         <?php endif; ?>
116                     </td>
117

```

```

118         <td>
119             <?php if($courseTutor): ?>
120                 <form method="POST" action="?action=remove_tutor" style="display: inline;" onsubm
121                     <input type="hidden" name="course_id" value="<?= $course['id'] ?>">
122                     <button type="submit" class="btn-danger" style="padding: 5px 10px; font-size:
123                         x Quitar
124                     </button>
125                 </form>
126             <?php else: ?>
127                 -
128             <?php endif; ?>
129         </td>
130     </tr>
131 <?php endforeach; ?>
132 </tbody>
133 </table>
134 </div>
135 </div>
136 </div>
137
138 <script>
139     let currentTutor = null;
140
141     document.getElementById('course_tutor').addEventListener('change', function() {
142         const courseId = this.value;
143         const teacherSelect = document.getElementById('teacher_tutor');
144         currentTutor = null;
145
146         teacherSelect.innerHTML = '<option value="">Cargando...</option>';
147
148         if (!courseId) {
149             teacherSelect.innerHTML = '<option value="">Primero seleccione un curso...</option>';
150             return;
151         }
152
153         fetch('?action=check_course_tutor&course_id=' + courseId)
154             .then(response => response.json())
155             .then(tutor => {
156                 if (tutor && tutor.tutor_name) {
157                     currentTutor = tutor.tutor_name;
158                 }
159             });
160
161         fetch('?action=get_course_teachers&course_id=' + courseId)
162             .then(response => response.json())
163             .then(data => {
164                 if (data.length === 0) {
165                     teacherSelect.innerHTML = '<option value="">No hay docentes asignados a este curso</option>';
166                 } else {
167                     teacherSelect.innerHTML = '<option value="">Seleccionar...</option>';
168                     data.forEach(teacher => {
169                         const option = document.createElement('option');
170                         option.value = teacher.teacher_id;
171                         option.textContent = teacher.teacher_name;
172                         teacherSelect.appendChild(option);
173                     });
174                 }
175             })
176             .catch(error => {

```

```

177         teacherSelect.innerHTML = '<option value="">Error al cargar docentes</option>';
178         console.error('Error:', error);
179     });
180 });
181
182 document.getElementById('tutorForm').addEventListener('submit', function(e) {
183     if (currentTutor) {
184         e.preventDefault();
185
186         const modal = document.createElement('div');
187         modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
188
189         const modalContent = document.createElement('div');
190         modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh.
191
192         modalContent.innerHTML = `
193             <h3 style="margin: 0 0 15px 0; color: #333;">■ Confirmar Cambio de Tutor</h3>
194             <p style="margin: 0 0 20px 0; color: #666;">
195                 El curso ya tiene como tutor a: <strong>${currentTutor}</strong>. <br>
196                 ¿Desea reemplazarlo por el nuevo docente?
197             </p>
198             <div style="display: flex; gap: 10px; justify-content: flex-end;">
199                 <button type="button" id="cancelBtn" style="padding: 10px 20px; background: #6c757d; color: white.
200                     Cancelar
201                 </button>
202                 <button type="button" id="confirmBtn" style="padding: 10px 20px; background: #007bff; color: whit.
203                     Sí, Reemplazar
204                 </button>
205             </div>
206         `;
207
208         modal.appendChild(modalContent);
209         document.body.appendChild(modal);
210
211         const form = this;
212
213         document.getElementById('confirmBtn').onclick = function() {
214             document.body.removeChild(modal);
215             currentTutor = null;
216             form.submit();
217         };
218
219         document.getElementById('cancelBtn').onclick = function() {
220             document.body.removeChild(modal);
221         };
222     }
223 });
224
225 function confirmRemoveTutor(event, tutorName, courseName) {
226     event.preventDefault();
227
228     const modal = document.createElement('div');
229     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
230
231     const modalContent = document.createElement('div');
232     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow.
233
234     modalContent.innerHTML = `
235         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Quitar Tutor</h3>

```

```

236         <p style="margin: 0 0 20px 0; color: #666;">
237             ¿Está seguro de quitar a <strong>${tutorName}</strong> como tutor del curso <strong>${.
238         </p>
239         <div style="display: flex; gap: 10px; justify-content: flex-end;">
240             <button type="button" id="cancelRemoveBtn" style="padding: 10px 20px; background: #6c757d; color: whi.
241                 Cancelar
242             </button>
243             <button type="button" id="confirmRemoveBtn" style="padding: 10px 20px; background: #dc3545; color: wh.
244                 Sí, Quitar
245             </button>
246         </div>
247     `;
248
249     modal.appendChild(modalContent);
250     document.body.appendChild(modal);
251
252     const form = event.target;
253
254     document.getElementById('confirmRemoveBtn').onclick = function() {
255         document.body.removeChild(modal);
256         form.submit();
257     };
258
259     document.getElementById('cancelRemoveBtn').onclick = function() {
260         document.body.removeChild(modal);
261     };
262
263     return false;
264 }
265 </script>
266 </body>
267 </html>

```

66. ArchImp/views/assignments/view_course.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Asignaciones del Curso - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; }
11         .navbar h1 { font-size: 24px; }
12         .navbar a { color: white; text-decoration: none; margin-left: 20px; }
13         .container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
14         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bottom: 20px; }
15         table { width: 100%; border-collapse: collapse; margin-top: 15px; }
16         th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
17         th { background: #f8f9fa; font-weight: bold; }
18         .course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
19         .tutor-box { background: #fff3cd; padding: 15px; border-radius: 4px; border-left: 4px solid #ffc107; margin-bottom: 20px; }
20         h2, h3 { color: #333; margin-bottom: 15px; }
21     </style>
22 </head>
23 <body>
24     <div class="navbar">
25         <h1>Asignaciones del Curso</h1>
26     </div>
27     <a href="?action=assignments">← Asignaciones</a>
28     <a href="?action=logout">Cerrar sesión</a>
29 </div>
30 </div>
31
32 <div class="container">
33     <div class="card">
34         <div class="course-info">
35             <h2><?= $course['name'] ?></h2>
36             <p><strong>Nivel:</strong> <?= $course['grade_level'] ?> | <strong>Paralelo:</strong> <?= $course['parallel'] ?>
37             <p><strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?> | <strong>Horario:</strong> <?= $course['shift_time'] ?>
38         </div>
39
40         <?php if($tutor): ?>
41             <div class="tutor-box">
42                 <h3>■■■ Docente Tutor</h3>
43                 <p><?= $tutor['last_name'] . ' ' . $tutor['first_name'] ?></p>
44                 <p style="font-size: 14px; color: #666;">Email: <?= $tutor['email'] ?></p>
45             </div>
46         <?php else: ?>
47             <div class="tutor-box">
48                 <p><strong>■ Este curso no tiene docente tutor asignado</strong></p>
49             </div>
50         <?php endif; ?>
51
52         <h3>Asignaturas y Docentes</h3>
53         <?php if(count($assignments) > 0): ?>
54             <table>
55                 <thead>
56                     <tr>
57                         <th>Asignatura</th>
58                         <th>Docente</th>
```

```

59         </tr>
60     </thead>
61     <tbody>
62         <?php foreach($assignments as $assignment): ?>
63             <tr>
64                 <td><?=$assignment['subject_name'] ?></td>
65                 <td>
66                     <?=$assignment['teacher_name'] ?>
67                     <?php if($assignment['is_tutor']): ?>
68                         <span style="color: #ffc107; font-weight: bold;">(Tutor)</span>
69                     <?php endif; ?>
70                 </td>
71             </tr>
72             <?php endforeach; ?>
73         </tbody>
74     </table>
75     <?php else: ?>
76         <p style="color: #666; margin-top: 15px;">No hay docentes asignados a este curso.</p>
77     <?php endif; ?>
78 </div>
79 </div>
80 </body>
81 </html>

```


67. ArchImp/views/attendance/calendar.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Calendario de Asistencia - EcuAsist</title>
8      <style>
9          .calendar-grid { display: grid; grid-template-columns: repeat(7, 1fr); gap: 6px; margin-top: 16px; }
10         .cal-header { text-align: center; font-size: 0.78rem; font-weight: 600; color: #555; padding: 8px; background: #.
11         .cal-day { border: 1px solid #e0e0e0; padding: 8px; min-height: 70px; border-radius: 6px; background: #fff; posi.
12         .cal-day.empty { background: #f8f9fa; border: none; }
13         .cal-day.today { background: #fff8e1; border-color: #ffc107; }
14         .cal-day.has-data { background: #f0faf0; border-color: #28a745; }
15         .cal-day.weekend { background: #fafafa; color: #bbb; }
16         .day-num { font-weight: 700; font-size: 0.9rem; margin-bottom: 4px; }
17         .day-summary { font-size: 0.72rem; color: #666; line-height: 1.4; }
18         .day-summary .ok { color: #28a745; font-weight: 600; }
19         .day-summary .no { color: #dc3545; font-weight: 600; }
20         .month-nav { display: flex; justify-content: space-between; align-items: center; margin-bottom: 16px; }
21         .month-nav h2 { font-size: 1.1rem; font-weight: 600; color: #333; }
22     </style>
23 </head>
24 <body>
25
26 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
27
28 <div class="breadcrumb">
29     <a href="?action=dashboard">■ Inicio</a> &rsquo;
30     <a href="?action=view_course_attendance">Asistencia</a> &rsquo;
31     Calendario
32 </div>
33
34 <div class="container">
35
36     <!-- Header -->
37     <div class="page-header teal">
38         <div class="ph-icon">■</div>
39     </div>
40     <h1>Calendario de Asistencia</h1>
41     <p>Vista mensual del registro de asistencia por curso</p>
42 </div>
43
44
45     <!-- Selector de curso -->
46     <div class="panel" style="margin-bottom: 16px;">
47         <div class="form-row">
48             <div class="form-group" style="flex: 2;">
49                 <label>Seleccionar Curso</label>
50                 <select id="course-select" class="form-control" onchange="loadCalendar()">
51                     <option value="">Seleccionar curso...</option>
52                     <?php foreach($courses as $course): ?>
53                         <option value="<? $course['id'] ?>"
54                             <? isset($_GET['course_id']) &amp; $_GET['course_id'] == $course['id'] ? 'selected'.
55                             <? htmlspecialchars($course['name']) ?> - <? ucfirst($course['shift_name']) ?>
56                         </option>
57                     <?php endforeach; ?>
58                 </select>
```

```

59         </div>
60     </div>
61 </div>
62
63 <?php if(isset($_GET['course_id'])): ?>
64
65 <!-- Navegación del mes -->
66 <div class="month-nav">
67     <a href="?action=attendance_calendar&course_id=<?=(int)$_GET['course_id'] ?&month=<?= html.
68     class="btn btn-outline">< Anterior</a>
69     <h2><?= htmlspecialchars($monthName) ?& <?= $year ?&</h2>
70     <a href="?action=attendance_calendar&course_id=<?=(int)$_GET['course_id'] ?&month=<?= html.
71     class="btn btn-outline">Siguiente →</a>
72 </div>
73
74 <!-- Calendario -->
75 <div class="panel" style="padding:16px;">
76     <div class="calendar-grid">
77         <?php foreach(['Dom', 'Lun', 'Mar', 'Mié', 'Jue', 'Vie', 'Sáb'] as $d): ?>
78             <div class="cal-header"><?= $d ?&</div>
79         <?php endforeach; ?>
80
81         <?php foreach($calendarDays as $day): ?>
82             <?php if(!$day['day']): ?>
83                 <div class="cal-day empty"></div>
84             <?php else: ?>
85                 <div class="cal-day <?= $day['classes'] ?&">
86                     <div class="day-num"><?= $day['day'] ?&</div>
87                     <?php if(isset($day['attendance'])): ?>
88                         <div class="day-summary">
89                             <span class="ok">✓ <?= $day['attendance']['presente'] ?&</span>
90                             &nbsp;
91                             <span class="no">✗ <?= $day['attendance']['ausente'] ?&</span>
92                         </div>
93                     <?php endif; ?>
94                 </div>
95             <?php endif; ?>
96         <?php endforeach; ?>
97     </div>
98
99 <!-- Leyenda -->
100 <div style="margin-top:16px;display:flex;gap:16px;flex-wrap:wrap;font-size:0.8rem;color:#666;">
101     <span><span style="display:inline-block;width:12px;height:12px;background:#f0faf0;border:1px solid .
102     <span><span style="display:inline-block;width:12px;height:12px;background:#fff8e1;border:1px solid .
103     <span><span style="display:inline-block;width:12px;height:12px;background:#fafafa;border:1px solid .
104 </div>
105 </div>
106
107 <?php else: ?>
108 <div class="empty-state">
109     <div class="icon">■</div>
110     <p>Selecione un curso para ver el calendario.</p>
111 </div>
112 <?php endif; ?>
113
114 </div>
115
116 <script>
117 function loadCalendar() {

```

```
118     const id = document.getElementById('course-select').value;
119     if (id) window.location.href = '?action=attendance_calendar&course_id=' + id;
120 }
121 </script>
122 </body>
123 </html>
124
```

68. ArchImp/views/attendance/course_view.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Ver Asistencia del Curso - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      Asistencia del Curso
16  </div>
17
18  <div class="container">
19
20      <?php if(empty($courses)): ?>
21          <div class="alert alert-warning">
22              ■ No tiene cursos asignados.
23              <?php if(Security::hasRole('docente')): ?>
24                  Contacte con la autoridad para que le asignen cursos.
25              <?php endif; ?>
26          </div>
27      <?php else: ?>
28
29          <!-- Header -->
30          <div class="page-header blue">
31              <div class="ph-icon">■</div>
32              <div>
33                  <h1>Asistencia del Curso</h1>
34                  <p>Consulta estadísticas de asistencia por curso y período</p>
35              </div>
36          </div>
37
38          <!-- Filtros -->
39          <div class="panel">
40              <form method="GET" id="filterForm">
41                  <input type="hidden" name="action" value="view_course_attendance">
42                  <div class="form-row" style="align-items:flex-end;">
43                      <div class="form-group" style="flex:2;">
44                          <label>Curso *</label>
45                          <select name="course_id" class="form-control" required>
46                              <option value="">Seleccionar curso...</option>
47                              <?php foreach($courses as $c):
48                                  $label = htmlspecialchars($c['name']);
49                                  if (!empty($c['is_tutor'])) $label .= ' ■ (Mi curso)';
50                              ?>
51                              <option value="<? $c['id'] ?>" <? isset($courseId) &amp;&amp; $courseId == $c[
52                                  <? $label ?>
53                              </option>
54                          <?php endforeach; ?>
55                      </select>
56                  </div>
57                  <div class="form-group">
58                      <label>Desde</label>
```

```

59         <input type="date" name="start_date" class="form-control"
60             value="&lt;?= htmlspecialchars($startDate ?? date('Y-m-01')) ?&gt;"&gt;
61     </div>&gt;
62     <div class="form-group"&gt;
63         <label>Hasta</label>
64         <input type="date" name="end_date" class="form-control"
65             value="&lt;?= htmlspecialchars($endDate ?? date('Y-m-d')) ?&gt;"&gt;
66     </div>&gt;
67     <div class="form-group" style="flex:0 0 auto;"&gt;
68         <label>&nbsp;&nbsp;&nbsp;</label>
69         <button type="submit" class="btn btn-primary" style="width:100%;"&gt;■ Consultar</button>
70     </div>&gt;
71 </div>&gt;
72 </form>&gt;
73 </div>&gt;
74
75 <?php if(isset($courseId) &amp;&amp; !empty($attendanceData)):
76     $totalPresente = array_sum(array_column($attendanceData, 'presente'));
77     $totalAusente = array_sum(array_column($attendanceData, 'ausente'));
78     $totalTardanza = array_sum(array_column($attendanceData, 'tardanza'));
79     $totalJustificado = array_sum(array_column($attendanceData, 'justificado'));
80     $totalRegistros = array_sum(array_column($attendanceData, 'total_registros'));
81     $pct = fn($v) => $totalRegistros &gt; 0 ? round($v / $totalRegistros * 100, 1) : 0;
82 ?&gt;
83
84 <!-- Stat cards --&gt;
85 <div style="display:flex;gap:12px;margin-bottom:20px;flex-wrap:wrap;"&gt;
86     <div class="stat-card gray"&gt;
87         <div class="number"&gt;&lt;?= $totalRegistros ?&gt;</div>
88         <div class="label"&gt;Total Registros</div>
89     </div>
90     <div class="stat-card green"&gt;
91         <div class="number"&gt;&lt;?= $totalPresente ?&gt;</div>
92         <div class="label"&gt;Presentes (&lt;?= $pct($totalPresente) ?&gt;%)</div>
93         <div class="progress-bar"&gt;<div class="progress-fill" style="width:&lt;?= $pct($totalPresente) ?&gt;.%</div>
94     </div>
95     <div class="stat-card red"&gt;
96         <div class="number"&gt;&lt;?= $totalAusente ?&gt;</div>
97         <div class="label"&gt;Ausentes (&lt;?= $pct($totalAusente) ?&gt;%)</div>
98         <div class="progress-bar"&gt;<div class="progress-fill" style="width:&lt;?= $pct($totalAusente) ?&gt;.%</div>
99     </div>
100    <div class="stat-card yellow"&gt;
101        <div class="number"&gt;&lt;?= $totalTardanza ?&gt;</div>
102        <div class="label"&gt;Tardanzas (&lt;?= $pct($totalTardanza) ?&gt;%)</div>
103        <div class="progress-bar"&gt;<div class="progress-fill" style="width:&lt;?= $pct($totalTardanza) ?&gt;.%</div>
104    </div>
105    <div class="stat-card teal"&gt;
106        <div class="number"&gt;&lt;?= $totalJustificado ?&gt;</div>
107        <div class="label"&gt;Justificados (&lt;?= $pct($totalJustificado) ?&gt;%)</div>
108        <div class="progress-bar"&gt;<div class="progress-fill" style="width:&lt;?= $pct($totalJustificado) ?&gt;.%</div>
109    </div>
110 </div>
111
112 <!-- Tabla estudiantes --&gt;
113 <div class="table-wrap"&gt;
114     <div class="table-info"&gt;
115         <span>■ Período: &lt;?= date('d/m/Y', strtotime($startDate)) ?&gt; al &lt;?= date('d/m/Y', strtotime($
116         <span>&lt;strong>&lt;?= count($attendanceData) ?&gt;</strong> estudiantes</span>
117     </div>

```

```

118         <table>
119             <thead>
120                 <tr>
121                     <th>#</th>
122                     <th>Estudiante</th>
123                     <th style="text-align:center;">Total</th>
124                     <th style="text-align:center;">Presentes</th>
125                     <th style="text-align:center;">Ausentes</th>
126                     <th style="text-align:center;">Tardanzas</th>
127                     <th style="text-align:center;">Justificados</th>
128                     <th style="text-align:center;">% Asistencia</th>
129                 </tr>
130             </thead>
131             <tbody>
132                 <?php $i = 1; foreach($attendanceData as $data):
133                     $pctRow = $data['total_registros'] > 0
134                         ? round(($data['presente'] / $data['total_registros']) * 100, 1) : 0;
135                     $pctColor = $pctRow >= 90 ? '#28a745' : ($pctRow >= 75 ? '#e6ac00' : '#dc3545');
136                     ?>
137                 <tr>
138                     <td style="color:#999;">&#x2013; $i++ </td>
139                     <td><strong>?<?php htmlspecialchars($data['student_name']) ?></strong></td>
140                     <td style="text-align:center;">?<?php $data['total_registros'] ?></td>
141                     <td style="text-align:center;">?<?php $data['presente'] .
142                     <td style="text-align:center;">?<?php $data['ausente'] .
143                     <td style="text-align:center;">?<?php $data['tardanza'] .
144                     <td style="text-align:center;">?<?php $data['justificado'] .
145                     <td style="text-align:center;font-weight:700;color:<?php $pctColor ?>;">?<?php $pctRow ?>
146                 </tr>
147                 <?php endforeach; ?>
148             </tbody>
149         </table>
150     </div>
151
152     <?php elseif(isset($courseId)): ?>
153         <div class="empty-state">
154             <div class="icon"><img alt="Empty state icon" /></div>
155             <p>No hay registros de asistencia para el período seleccionado.</p>
156         </div>
157     <?php else: ?>
158         <div class="empty-state">
159             <div class="icon"><img alt="Empty state icon" /></div>
160             <p>Seleccione un curso y período para ver la asistencia.</p>
161         </div>
162     <?php endif; ?>
163
164 <?php endif; ?>
165
166 </div>
167 </body>
168 </html>
169

```

69. ArchImp/views/attendance/my_attendance.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mi Asistencia - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      Mi Asistencia
16  </div>
17
18  <div class="container">
19
20  <?php
21  $total      = count($attendances);
22  $presente   = count(array_filter($attendances, fn($a) => $a['status'] === 'presente'));
23  $ausente    = count(array_filter($attendances, fn($a) => $a['status'] === 'ausente'));
24  $tardanza   = count(array_filter($attendances, fn($a) => $a['status'] === 'tardanza'));
25  $pctAsist   = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
26  $pct        = fn($v) => $total > 0 ? round($v / $total * 100, 1) : 0;
27  ?>
28
29  <!-- Header -->
30  <div class="page-header green">
31      <div class="ph-icon">■</div>
32      <div>
33          <h1>Mi Asistencia</h1>
34          <p>Historial completo de tu asistencia escolar</p>
35      </div>
36      <div class="ph-actions">
37          <a href="?action=my_justifications" class="btn btn-outline" style="color:#fff;border-color:rgba(255,255,255,0.5);">
38              ■ Mis Justificaciones
39          </a>
40      </div>
41  </div>
42
43  <!-- Stat cards -->
44  <div style="display:flex;gap:12px;margin-bottom:20px;flex-wrap:wrap;">
45      <div class="stat-card gray">
46          <div class="number"><?= $total ?></div>
47          <div class="label">Total Registros</div>
48      </div>
49      <div class="stat-card green">
50          <div class="number"><?= $pctAsist ?>%</div>
51          <div class="label">Asistencia (<?= $presente ?> presentes)</div>
52          <div class="progress-bar"><div class="progress-fill" style="width:<?= $pctAsist ?>%;background-color:#28a745;"></div>
53      </div>
54      <div class="stat-card red">
55          <div class="number"><?= $ausente ?></div>
56          <div class="label">Ausencias (<?= $pct($ausente) ?>%)</div>
57          <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct($ausente) ?>%;background-color:#dc3545;"></div>
58      </div>
59  </div>
```

```

59         <div class="stat-card yellow">
60             <div class="number"><? $tardanza ?></div>
61             <div class="label">Tardanzas (<? $pct($tardanza) ?>%</div>
62             <div class="progress-bar"><div class="progress-fill" style="width:<? $pct($tardanza) ?>%<
63         </div>
64     </div>
65
66     <!-- Tabla -->
67     <div class="table-wrap">
68         <div class="table-info">
69             <span>■ <strong><? $total ?></strong> registros</span>
70         </div>
71         <?php if(empty($attendances)): ?>
72         <div class="empty-state">
73             <div class="icon">■</div>
74             <p>No hay registros de asistencia.</p>
75         </div>
76         <?php else: ?>
77         <table>
78             <thead>
79                 <tr>
80                     <th>Fecha</th>
81                     <th>Curso</th>
82                     <th>Asignatura</th>
83                     <th>Jornada</th>
84                     <th>Hora</th>
85                     <th>Estado</th>
86                     <th>Observación</th>
87                     <th>Acción</th>
88                 </tr>
89             </thead>
90             <tbody>
91                 <?php foreach($attendances as $att):
92                     $badgeClass = match($att['status']) {
93                         'presente'     => 'badge-green',
94                         'ausente'      => 'badge-red',
95                         'tardanza'     => 'badge-yellow',
96                         'justificado'  => 'badge-teal',
97                         default        => 'badge-gray'
98                     };
99                     $label = match($att['status']) {
100                         'presente'     => 'Presente',
101                         'ausente'      => 'Ausente',
102                         'tardanza'     => 'Tardanza',
103                         'justificado'  => 'Justificado',
104                         default        => ucfirst($att['status'])
105                     };
106                 <?>
107                 <tr>
108                     <td><? date('d/m/Y', strtotime($att['date'])) ?></td>
109                     <td><? htmlspecialchars($att['course_name']) ?></td>
110                     <td><? htmlspecialchars($att['subject_name']) ?></td>
111                     <td style="color:#666;"><? ucfirst($att['shift_name']) ?></td>
112                     <td style="white-space:nowrap;"><? htmlspecialchars($att['hour_period']) ?></td>
113                     <td><span class="badge <? $badgeClass ?>"><? $label ?></span></td>
114                     <td style="color:#888;font-size:0.82rem;"><? htmlspecialchars($att['observation']) ?? '-' ?>
115                 </td>
116                 <?php if($att['status'] === 'ausente'): ?>
117                     <a href="?action=submit_justification&attendance_id=<? $att['id'] ?>"

```



```
118         class="btn btn-warning btn-sm"&gt;■ Justificar&lt;/a&gt;
119     &lt;?php else: ?&gt;
120         &lt;span style="color:#ccc;"&gt;—&lt;/span&gt;
121     &lt;?php endif; ?&gt;
122     &lt;/td&gt;
123 &lt;/tr&gt;
124 &lt;?php endforeach; ?&gt;
125 &lt;/tbody&gt;
126 &lt;/table&gt;
127 &lt;?php endif; ?&gt;
128 &lt;/div&gt;
129
130 &lt;/div&gt;
131 &lt;/body&gt;
132 &lt;/html&gt;
133
```

70. ArchImp/views/attendance/register.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Registrar Asistencia - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsaquo;
15      <a href="?action=attendance_view">Asistencia</a> &rsaquo;
16      Registrar Asistencia
17  </div>
18
19  <div class="container">
20
21      <?php if(isset($_GET['success'])): ?>
22          <div class="alert alert-success">✓ Asistencia registrada correctamente</div>
23      <?php endif; ?>
24      <?php if(isset($_GET['error'])): ?>
25          <?php if($_GET['error'] == 'future'): ?>
26              <div class="alert alert-danger">✗ No se puede registrar asistencia de fechas futuras</div>
27          <?php elseif($_GET['error'] == 'toolate'): ?>
28              <div class="alert alert-danger">✗ Solo se puede registrar asistencia de hasta 48 horas hábiles atrás&am
29          <?php endif; ?>
30      <?php endif; ?>
31
32      <!-- Header -->
33      <div class="page-header green">
34          <div class="ph-icon">■</div>
35          <div>
36              <h1>Registrar Asistencia</h1>
37              <p>Selecciona tu clase del día y toma asistencia de los estudiantes</p>
38          </div>
39      </div>
40
41      <!-- Clases del día -->
42      <div class="panel">
43          <h3>■ Mis Clases de Hoy</h3>
44          <div class="alert alert-warning" style="margin-bottom:12px;">
45              ■ <strong>Importante:</strong> Puede registrar asistencia de hoy o corregir registros anteriores.
46              <strong><? = date('d/m/Y', strtotime($maxEditDate)) ?></strong>.
47          </div>
48
49          <?php if(empty($todayClasses)): ?>
50              <div class="empty-state">
51                  <div class="icon">■</div>
52                  <p>No tienes clases programadas para hoy según tu horario.</p>
53              </div>
54          <?php else: ?>
55              <div id="classes-list" style="display:grid;grid-template-columns:repeat(auto-fill,minmax(280px,1fr));gap:.
56                  <?php foreach($todayClasses as $class):
57                      $jsonCourse = json_encode($class['course_name'], JSON_HEX_TAG | JSON_HEX_APOS);
58                      $jsonSubject = json_encode($class['subject_name'], JSON_HEX_TAG | JSON_HEX_APOS);
```

```

59         ?&gt;
60         &lt;div class="class-card" onclick='selectClass(event, &lt;?=$class["id"] ?&gt;; &lt;?=$jsonCourse ?&am
61             &lt;div style="font-size:1.5rem;margin-bottom:6px;"&gt;■&lt;/div&gt;
62             &lt;div style="font-weight:700;color:#007bff;font-size:0.95rem;margin-bottom:4px;"&gt;
63                 &lt;?=$class['period_number'] ?&gt;;ra hora - &lt;?=$htmlspecialchars($class['subject_name']) ?&a
64             &lt;/div&gt;
65             &lt;div style="font-size:0.83rem;color:#666;"&gt;Curso: &lt;?=$htmlspecialchars($class['course_name']
66             &lt;div style="margin-top:10px;"&gt;
67                 &lt;span class="btn btn-primary btn-sm"&gt;Tomar Asistencia →&lt;/span&gt;
68             &lt;/div&gt;
69         &lt;/div&gt;
70         &lt;?php endforeach; ?&gt;
71     &lt;/div&gt;
72     &lt;?php endif; ?&gt;
73 &lt;/div&gt;
74
75 &lt;!-- Selector de fecha --&gt;
76 &lt;div class="panel" id="date-selector" style="display:none;"&gt;
77     &lt;h3&gt;■ Seleccionar Fecha&lt;/h3&gt;
78     &lt;form id="config-form"&gt;
79         &lt;input type="hidden" id="schedule_id" name="schedule_id"&gt;
80         &lt;div class="form-row"&gt;
81             &lt;div class="form-group"&gt;
82                 &lt;label&gt;Clase Seleccionada&lt;/label&gt;
83                 &lt;input type="text" id="selected_class_info" class="form-control" readonly style="background:#f8f9
84             &lt;/div&gt;
85             &lt;div class="form-group"&gt;
86                 &lt;label&gt;Fecha&lt;/label&gt;
87                 &lt;input type="date" id="date" name="date" class="form-control"
88                     value="&lt;?=$date('Y-m-d') ?&gt;"
89                     max="&lt;?=$date('Y-m-d') ?&gt;"
90                     min="&lt;?=$minDate ?&gt;" required&gt;
91                 &lt;div class="form-hint"&gt;■ La asistencia de hoy podrá editarse hasta el &lt;?=$date('d/m/Y', str
92             &lt;/div&gt;
93         &lt;/div&gt;
94         &lt;button type="button" class="btn btn-primary" onclick="loadStudents()"&gt;■ Cargar Estudiantes&lt;/button
95     &lt;/form&gt;
96 &lt;/div&gt;
97
98 &lt;!-- Lista de estudiantes --&gt;
99 &lt;form method="POST" id="student-list" style="display:none;"&gt;
100     &lt;input type="hidden" name="schedule_id" id="schedule_id_hidden"&gt;
101     &lt;input type="hidden" name="date" id="date_hidden"&gt;
102
103     &lt;div class="panel"&gt;
104         &lt;h3&gt;■ Lista de Estudiantes&lt;/h3&gt;
105         &lt;div class="table-wrap" style="margin-top:8px;"&gt;
106             &lt;table id="students-table"&gt;
107                 &lt;thead&gt;
108                     &lt;tr&gt;
109                         &lt;th&gt;#&lt;/th&gt;
110                         &lt;th&gt;Apellidos y Nombres&lt;/th&gt;
111                         &lt;th&gt;Cédula&lt;/th&gt;
112                         &lt;th&gt;Estado&lt;/th&gt;
113                         &lt;th&gt;Observación&lt;/th&gt;
114                     &lt;/tr&gt;
115                 &lt;/thead&gt;
116                 &lt;tbody&gt;&lt;/tbody&gt;
117             &lt;/table&gt;

```

```

118         </div>;
119         <div style="margin-top:16px;">
120             <button type="submit" class="btn btn-success btn-lg">■ Guardar Asistencia</button>
121         </div>;
122     </div>;
123 </form>;
124
125 </div>;
126
127 <style>
128 .class-card {
129     background: #fff;
130     border: 2px solid #e0e0e0;
131     border-radius: 8px;
132     padding: 16px;
133     cursor: pointer;
134     transition: border-color 0.2s, box-shadow 0.2s;
135 }
136 .class-card:hover {
137     border-color: #007bff;
138     box-shadow: 0 4px 12px rgba(0,123,255,0.15);
139 }
140 .class-card.selected {
141     border-color: #007bff;
142     background: #f0f7ff;
143 }
144 </style>
145
146 <script>
147 const minDateAllowed = '<?=? $minDate ?>';
148 let selectedScheduleId = null;
149
150 function selectClass(event, scheduleId, courseName, subjectName, periodNumber) {
151     selectedScheduleId = scheduleId;
152     document.querySelectorAll('.class-card').forEach(c => c.classList.remove('selected'));
153     event.currentTarget.classList.add('selected');
154     document.getElementById('date-selector').style.display = 'block';
155     document.getElementById('schedule_id').value = scheduleId;
156     document.getElementById('selected_class_info').value = periodNumber + 'ra hora - ' + subjectName + ' (' + courseName.
157     document.getElementById('student-list').style.display = 'none';
158     document.getElementById('date-selector').scrollIntoView({ behavior: 'smooth' });
159 }
160
161 document.getElementById('date') && document.getElementById('date').addEventListener('change', function() {
162     const today = new Date().toISOString().split('T')[0];
163     if (this.value > today) { alert('No se puede seleccionar una fecha futura'); this.value = today; return; }
164     if (this.value < minDateAllowed) { alert('Solo puede registrar asistencia de hasta 48 horas hábiles atrás'); this.
165 });
166
167 function loadStudents() {
168     const scheduleId = document.getElementById('schedule_id').value;
169     const date = document.getElementById('date').value;
170     if (!scheduleId || !date) { alert('Complete todos los campos'); return; }
171
172     document.getElementById('schedule_id_hidden').value = scheduleId;
173     document.getElementById('date_hidden').value = date;
174
175     fetch('?action=get_schedule_info&schedule_id=' + scheduleId)
176         .then(r => r.json())

```

```

177     .then(scheduleData => {
178         const formData = new FormData();
179         formData.append('course_id', scheduleData.course_id);
180         return fetch('?action=get_students', { method: 'POST', body: formData })
181             .then(r => r.json())
182             .then(students => {
183                 if (!students.length) { alert('No hay estudiantes matriculados en este curso'); return; }
184                 return fetch('?action=get_existing_attendance&schedule_id=' + scheduleId + '&date=' + date)
185                     .then(r => r.json())
186                     .then(existing => {
187                         const existingMap = {};
188                         existing.forEach(a => { existingMap[a.student_id] = { status: a.status, observation: a.ob.
189                         const tbody = document.querySelector('#students-table tbody');
190                         tbody.innerHTML = '';
191                         const statuses = ['presente', 'ausente', 'tardanza', 'justificado'];
192                         students.forEach((student, idx) => {
193                             const cur = existingMap[student.id] || { status: 'presente', observation: '' };
194                             const opts = statuses.map(s => `<option value="${s}" ${cur.status===s?'selected':}`.
195                             tbody.innerHTML += `<tr>
196                                 <td style="color:#999;">${idx+1}</td>
197                                 <td><strong>${student.last_name} ${student.first_name}</strong>&
198                                 <td style="color:#666;">${student.dni||'-'}</td>
199                                 <td><select name="status_${student.id}" style="padding:5px 8px;border:1px s.
200                                 <td><input type="text" name="obs_${student.id}" value="${cur.observation}" .
201                             </tr>`;
202                         });
203                         document.getElementById('student-list').style.display = 'block';
204                         document.getElementById('student-list').scrollIntoView({ behavior: 'smooth' });
205                     });
206                 });
207             })
208         .catch(err => console.error('Error:', err));
209     }
210 </script>
211
212 </body>
213 </html>
214

```

71. ArchImp/views/attendance/view.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Ver Asistencias - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      Asistencia &rsquo;
16      Ver Asistencias
17  </div>
18
19  <div class="container">
20
21      <!-- Header -->
22      <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
23          <div class="ph-icon">■</div>
24          <div>
25              <h1>Ver Asistencias</h1>
26              <p>Consulta el registro de asistencia por curso y fecha</p>
27          </div>
28      </div>
29
30      <!-- Filtros -->
31      <div class="panel" style="margin-bottom:20px;">
32          <h3 style="margin-bottom:16px;font-size:.95rem;color:#555;">■ Filtros de Búsqueda</h3>
33          <form method="POST" id="filterForm">
34              <div class="form-row" style="align-items:flex-end;">
35                  <div class="form-group" style="flex:2;">
36                      <label>Curso</label>
37                      <select name="course_id" class="form-control" required>
38                          <option value="">Seleccionar curso...</option>
39                          <?php foreach($courses as $course): ?>
40                              <option value="<? $course['id'] ?>"
41                                  <? (isset($_POST['course_id']) &amp; $_POST['course_id'] == $course['id']) ? 's.' : ''>
42                                  <? htmlspecialchars($course['name']) ?>
43                                  - <? ucfirst($course['shift_name']) ?>
44                              </option>
45                          <?php endforeach; ?>
46                      </select>
47                  </div>
48                  <div class="form-group" style="flex:1;">
49                      <label>Fecha</label>
50                      <input type="date" name="date" class="form-control"
51                          value="<? (isset($_POST['date']) ? htmlspecialchars($_POST['date']) : date('Y-m-d')) ?>"
52                          max="<? date('Y-m-d') ?>" required>
53                  </div>
54                  <div class="form-group" style="flex:0 0 auto;">
55                      <button type="submit" class="btn btn-primary">■ Buscar</button>
56                  </div>
57              </div>
58          </form>
```

```

59     </div>
60
61     <?php if($_SERVER['REQUEST_METHOD'] === 'POST'): ?>
62
63         <?php if(empty($attendances)): ?>
64             <!-- Sin resultados -->
65             <div class="empty-state">
66                 <div class="icon">■</div>
67                 <p>No se encontraron registros de asistencia para los filtros seleccionados.</p>
68                 <small style="color:#aaa;">Verifica que el docente haya registrado asistencia para este curso y fe.
69             </div>
70
71         <?php else:
72             /* ---- Calcular estadísticas ---- */
73             $stats = ['presente'=>0,'ausente'=>0,'tardanza'=>0,'justificado'=>0];
74             $students = [];
75             foreach($attendances as $att) {
76                 $s = $att['status'];
77                 if(isset($stats[$s])) $stats[$s]++;
78                 $students[$att['last_name'].' '.$att['first_name']] = true;
79             }
80             $total = count($attendances);
81             $totalStu = count($students);
82             $pctPresente = $total > 0 ? round(($stats['presente'] + $stats['justificado']) / $total * 100) : 0;
83
84             $selectedCourse = '';
85             foreach($courses as $c) {
86                 if($c['id'] == $_POST['course_id']) {
87                     $selectedCourse = $c['name'] . ' - ' . ucfirst($c['shift_name']);
88                     break;
89                 }
90             }
91             ?>
92
93         <!-- Stat cards -->
94         <div style="display:grid;grid-template-columns:repeat(5,1fr);gap:16px;margin-bottom:20px;">
95             <div class="stat-card">
96                 <div class="stat-icon" style="background:#e3f2fd;">■</div>
97                 <div class="stat-value"><?=$total ?></div>
98                 <div class="stat-label">Total registros</div>
99             </div>
100            <div class="stat-card">
101                <div class="stat-icon" style="background:#e8f5e9;">■</div>
102                <div class="stat-value" style="color:#2e7d32;"><?=$stats['presente'] ?></div>
103                <div class="stat-label">Presentes</div>
104            </div>
105            <div class="stat-card">
106                <div class="stat-icon" style="background:#ffebee;">■</div>
107                <div class="stat-value" style="color:#c62828;"><?=$stats['ausente'] ?></div>
108                <div class="stat-label">Ausentes</div>
109            </div>
110            <div class="stat-card">
111                <div class="stat-icon" style="background:#fffde7;">■</div>
112                <div class="stat-value" style="color:#f57f17;"><?=$stats['tardanza'] ?></div>
113                <div class="stat-label">Tardanzas</div>
114            </div>
115            <div class="stat-card">
116                <div class="stat-icon" style="background:#e0f7fa;">■</div>
117                <div class="stat-value" style="color:#00838f;"><?=$stats['justificado'] ?></div>

```

```

118         <div class="stat-label">Justificados</div>
119     </div>
120 </div>
121
122 <!-- Barra de asistencia global -->
123 <div class="panel" style="margin-bottom:20px;">
124     <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:8px;">
125         <span style="font-size:.9rem;color:#555;">
126             ■ <strong><? htmlspecialchars($selectedCourse) ?></strong>
127             &nbsp;|&nbsp; ■ <? date('d/m/Y', strtotime($_POST['date'])) ?>
128             &nbsp;|&nbsp; ■ <? $totalStu ?> estudiante<? $totalStu!=1?'s':'' ?>
129         </span>
130         <span style="font-weight:700;font-size:1.1rem;color:<? $pctPresente>=90?'#2e7d32':($pctPresent.
131             <? $pctPresente ?>% asistencia efectiva
132         </span>
133     </div>
134     <div style="background:#f0f0f0;border-radius:6px;height:10px;overflow:hidden;">
135         <div style="height:100%;border-radius:6px;width:<? $pctPresente ?>%;">
136             background:<? $pctPresente>=90?'#4caf50':($pctPresente>=75?'#ff9800':'#f44336') ?>;
137             transition:width .4s;">
138         </div>
139     </div>
140 </div>
141
142 <!-- Tabla de resultados -->
143 <div class="table-wrap">
144     <div class="table-info">
145         <span>■ <strong>Registros de Asistencia</strong> - <? $total ?></span>
146         <div style="display:flex;gap:8px;align-items:center;">
147             <input type="text" id="searchTable" class="form-control"
148                 placeholder="■ Buscar estudiante..."
149                 style="width:200px;font-size:13px;"
150                 oninput="filterTable(this.value)">
151         </div>
152     </div>
153     <table id="attendanceTable">
154         <thead>
155             <tr>
156                 <th>#</th>
157                 <th>Estudiante</th>
158                 <th>Asignatura</th>
159                 <th>Docente</th>
160                 <th style="text-align:center;">Hora</th>
161                 <th style="text-align:center;">Estado</th>
162                 <th>Observación</th>
163             </tr>
164         </thead>
165         <tbody>
166             <?php
167             $i = 1;
168             foreach($attendances as $att):
169                 $s = $att['status'];
170                 $badgeMap = [
171                     'presente' => 'badge-green',
172                     'ausente' => 'badge-red',
173                     'tardanza' => 'badge-yellow',
174                     'justificado' => 'badge-teal',
175                 ];
176                 $labelMap = [

```



```

177         'presente'      => '✓ Presente',
178         'ausente'       => 'X Ausente',
179         'tardanza'      => '■ Tardanza',
180         'justificado'   => '■ Justificado',
181     ];
182     $badge = $badgeMap[$s] ?? 'badge-gray';
183     $label = $labelMap[$s] ?? ucfirst($s);
184     ?&gt;
185     &lt;tr&gt;
186         &lt;td style="color:#999;font-size:12px;"&gt;&lt;?=$i++ ?&gt;&lt;/td&gt;
187         &lt;td&gt;
188             &lt;strong&gt;&lt;?=$htmlspecialchars($att['last_name']) . ' ' . $att['first_name']) ?&gt;&amp;
189         &lt;/td&gt;
190         &lt;td style="color:#555;"&gt;&lt;?=$htmlspecialchars($att['subject_name']) ?&gt;&lt;/td&gt;
191         &lt;td style="font-size:13px;color:#666;"&gt;&lt;?=$htmlspecialchars($att['teacher_name']) ?&gt;.
192         &lt;td style="text-align:center;"&gt;
193             &lt;span style="font-size:13px;background:#f0f0f0;padding:3px 8px;border-radius:4px;"&gt;
194                 &lt;?=$htmlspecialchars($att['hour_period']) ?&gt;
195             &lt;/span&gt;
196         &lt;/td&gt;
197         &lt;td style="text-align:center;"&gt;
198             &lt;span class="badge &lt;?=$badge ?&gt;"&gt;&lt;?=$label ?&gt;&lt;/span&gt;
199         &lt;/td&gt;
200         &lt;td style="font-size:13px;color:#777;"&gt;
201             &lt;?=$att['observation'] ? $htmlspecialchars($att['observation']) : '&lt;span style="color:.
202         &lt;/td&gt;
203     &lt;/tr&gt;
204     &lt;?php endforeach; ?&gt;
205     &lt;/tbody&gt;
206     &lt;/table&gt;
207     &lt;/div&gt;
208
209     &lt;!-- Leyenda --&gt;
210     &lt;div style="margin-top:12px;display:flex;gap:12px;flex-wrap:wrap;"&gt;
211         &lt;span class="badge badge-green"&gt;✓ Presente&lt;/span&gt;
212         &lt;span class="badge badge-red"&gt;X Ausente&lt;/span&gt;
213         &lt;span class="badge badge-yellow"&gt;■ Tardanza&lt;/span&gt;
214         &lt;span class="badge badge-teal"&gt;■ Justificado&lt;/span&gt;
215     &lt;/div&gt;
216
217     &lt;?php endif; ?&gt;
218     &lt;?php endif; ?&gt;
219
220 &lt;/div&gt;
221
222 &lt;script&gt;
223 function filterTable(q) {
224     q = q.toLowerCase();
225     const rows = document.querySelectorAll('#attendanceTable tbody tr');
226     rows.forEach(r => {
227         const name = r.cells[1] ? r.cells[1].textContent.toLowerCase() : '';
228         r.style.display = name.includes(q) ? '' : 'none';
229     });
230 }
231 &lt;/script&gt;
232
233 &lt;/body&gt;
234 &lt;/html&gt;

```

72. ArchImp/views/auth/forgot.php

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Recuperar Contraseña - EcuAsist</title>
6      <style>
7          * { margin: 0; padding: 0; box-sizing: border-box; }
8          body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center; align-items: center; }
9          .container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); width: 100%; }
10         h2 { margin-bottom: 20px; color: #333; }
11         input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
12         button { width: 100%; padding: 12px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointer; }
13         button:hover { background: #0056b3; }
14         .success { color: green; margin-bottom: 10px; }
15         a { color: #007bff; }
16     </style>
17 </head>
18 <body>
19     <div class="container">
20         <h2>Recuperar Contraseña</h2>
21         <?php if(isset($message)): ?>
22             <p class="success"><?php echo $message ?></p>
23         <?php endif; ?>
24
25         <form method="POST">
26             <input type="hidden" name="csrf_token" value="<?php echo Security::generateToken() ?>">
27             <input type="email" name="email" placeholder="Correo electrónico" required>
28             <button type="submit">Enviar enlace</button>
29         </form>
30
31         <p style="margin-top: 15px;"><a href="?action=login">Volver al login</a></p>
32     </div>
33 </body>
34 </html>
```

73. ArchImp/views/auth/login.php

```
1  <?php
2  if(isset($_SESSION['user_id'])) {
3      header('Location: ?action=dashboard');
4      exit;
5  }
6  ?>
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10     <meta charset="UTF-8">
11     <meta name="viewport" content="width=device-width, initial-scale=1.0">
12     <title>Iniciar Sesión - EcuAsist 2026</title>
13     <style>
14         * { box-sizing: border-box; margin: 0; padding: 0; }
15         body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
16             background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
17             min-height: 100vh; display: flex; align-items: center; justify-content: center; }
18         .login-card { background: #fff; border-radius: 14px; box-shadow: 0 12px 40px rgba(0,0,0,0.2);
19             max-width: 420px; width: 100%; padding: 40px; }
20         .login-logo { text-align: center; margin-bottom: 28px; }
21         .login-logo .icon { font-size: 3.5rem; margin-bottom: 12px; }
22         .login-logo h2 { font-size: 1.5rem; font-weight: 700; color: #333; }
23         .login-logo p { color: #888; font-size: 0.88rem; margin-top: 4px; }
24         .form-group { margin-bottom: 18px; }
25         .form-group label { display: block; margin-bottom: 6px; font-size: 0.85rem; font-weight: 600; color: #444; }
26         .form-group input { width: 100%; padding: 10px 14px; border: 1px solid #ddd; border-radius: 8px;
27             font-size: 0.9rem; transition: border-color 0.2s; }
28         .form-group input:focus { border-color: #667eea; outline: none; box-shadow: 0 0 3px rgba(102,126,234,0.15); }
29         .btn-login { width: 100%; padding: 12px; background: linear-gradient(135deg, #667eea, #764ba2);
30             color: #fff; border: none; border-radius: 8px; font-size: 0.95rem; font-weight: 600;
31             cursor: pointer; transition: opacity 0.2s; margin-top: 4px; }
32         .btn-login:hover { opacity: 0.9; }
33         .alert { padding: 10px 14px; border-radius: 8px; margin-bottom: 18px; font-size: 0.85rem; }
34         .alert-danger { background: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }
35         .alert-success { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
36         .login-footer { text-align: center; margin-top: 20px; }
37         .login-footer a { color: #667eea; text-decoration: none; font-size: 0.85rem; }
38         .login-footer a:hover { text-decoration: underline; }
39         .divider { border: none; border-top: 1px solid #f0f0f0; margin: 20px 0; }
40         .demo-hint { background: #f8f9fa; border-radius: 8px; padding: 10px 14px; font-size: 0.78rem; color: #888; text-align: center; }
41         .demo-hint strong { color: #555; }
42     </style>
43 </head>
44 <body>
45     <div class="login-card">
46         <div class="login-logo">
47             <div class="icon">■</div>
48             <h2>EcuAsist 2026</h2>
49             <p>Sistema de Asistencia Escolar</p>
50         </div>
51
52         <?php if(isset($error)): ?>
53             <div class="alert alert-danger">■ <?php echo htmlspecialchars($error) ?></div>
54         <?php endif; ?>
55
56         <?php if(isset($_GET['registered'])): ?>
57             <div class="alert alert-success">✓ Registro exitoso. Ahora puede iniciar sesión.</div>
58         <?php endif; ?>
```

```

59
60     <?php if(isset($_GET['reset'])): ?>
61     <div class="alert alert-success">✓ Contraseña restablecida correctamente.</div>
62     <?php endif; ?>
63
64     <form method="POST">
65         <?php Security::generateToken(); ?>
66         <input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">
67
68         <div class="form-group">
69             <label>■ Usuario o Correo Electrónico</label>
70             <input type="text" name="email" required autofocus placeholder="Ingrese usuario o email">
71         </div>
72
73         <div class="form-group">
74             <label>■ Contraseña</label>
75             <input type="password" name="password" required placeholder="Ingrese contraseña">
76         </div>
77
78         <button type="submit" class="btn-login">■ Iniciar Sesión</button>
79     </form>
80
81     <div class="login-footer" style="margin-top:16px;">
82         <a href="?action=forgot">¿Olvidó su contraseña?</a>
83     </div>
84
85     <hr class="divider">
86
87     <div class="demo-hint">
88         ■■ Usuario de prueba: <strong>admin</strong> / <strong>password</strong>
89     </div>
90 </div>
91 </body>
92 </html>
93

```

74. ArchImp/views/auth/register.php

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Registro - EcuAsist</title>
6      <style>
7          * { margin: 0; padding: 0; box-sizing: border-box; }
8          body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center; align-items: center; }
9          .container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); width: 100%; }
10         h2 { margin-bottom: 20px; color: #333; }
11         input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
12         button { width: 100%; padding: 12px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; }
13         button:hover { background: #218838; }
14         a { color: #007bff; }
15     </style>
16 </head>
17 <body>
18     <div class="container">
19         <h2>Registro de Usuario</h2>
20         <form method="POST">
21             <input type="hidden" name="csrf_token" value="<?= Security::generateToken() ?>">
22             <input type="text" name="username" placeholder="Usuario" required>
23             <input type="email" name="email" placeholder="Correo electrónico" required>
24             <input type="password" name="password" placeholder="Contraseña" required>
25             <input type="text" name="first_name" placeholder="Nombres" required>
26             <input type="text" name="last_name" placeholder="Apellidos" required>
27             <input type="text" name="dni" placeholder="Cédula (opcional)">
28             <input type="text" name="phone" placeholder="Teléfono (opcional)">
29             <button type="submit">Registrarse</button>
30         </form>
31         <p style="margin-top: 15px;"><a href="?action=login">Ya tengo cuenta</a></p>
32     </div>
33 </body>
34 </html>
```

75. ArchImp/views/auth/reset.php

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Nueva Contraseña - EcuAsist</title>
6      <style>
7          * { margin: 0; padding: 0; box-sizing: border-box; }
8          body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center; align-items: center; height: 100vh; }
9          .container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); width: 400px; }
10         h2 { margin-bottom: 20px; color: #333; }
11         input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
12         button { width: 100%; padding: 12px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; }
13         button:hover { background: #218838; }
14     </style>
15 </head>
16 <body>
17     <div class="container">
18         <h2>Nueva Contraseña</h2>
19         <form method="POST">
20             <input type="hidden" name="csrf_token" value="<?= Security::generateToken() ?>">
21             <input type="password" name="password" placeholder="Nueva contraseña" required>
22             <input type="password" name="password_confirm" placeholder="Confirmar contraseña" required>
23             <button type="submit">Cambiar contraseña</button>
24         </form>
25     </div>
26 </body>
27 </html>
```

76. ArchImp/views/dashboard/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Dashboard - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      ■ Inicio &rsquo; Dashboard
15  </div>
16
17  <div class="container">
18
19      <!-- Header de bienvenida -->
20      <div class="page-header" style="background: linear-gradient(135deg, #667eea, #764ba2);">
21          <div class="ph-icon">■</div>
22          <div>
23              <h1>Bienvenido, <?php echo htmlspecialchars($_SESSION['first_name'] . ' ' . $_SESSION['last_name']) ?>
24              <p>
25                  <?php foreach($_SESSION['roles'] as $role): ?>
26                      <span class="badge badge-blue"><?php echo ucfirst($role) ?></span>
27                  <?php endforeach; ?>
28              </p>
29          </div>
30      </div>
31
32      <!-- ■■ Stats: Autoridad ■■ -->
33      <?php if(Security::hasRole('autoridad') && isset($stats['total_students'])): ?>
34          <div class="stats-row">
35              <div class="stat-card blue">
36                  <div class="number"><?php echo $stats['total_students'] ?></div>
37                  <div class="label">Estudiantes</div>
38              </div>
39              <div class="stat-card green">
40                  <div class="number"><?php echo $stats['total_teachers'] ?></div>
41                  <div class="label">Docentes</div>
42              </div>
43              <div class="stat-card orange">
44                  <div class="number"><?php echo $stats['total_courses'] ?></div>
45                  <div class="label">Cursos</div>
46              </div>
47              <?php if(!empty($stats['today_attendance']['total']) && $stats['today_attendance']['total'] > 0): .
48                  <div class="stat-card teal">
49                      <div class="number"><?php echo round(($stats['today_attendance']['presente'] / $stats['today_attendance']['total']) * 100) ?>%</div>
50                      <div class="label">Asistencia Hoy</div>
51                      <div class="progress-bar">
52                          <div class="progress-fill" style="width:<?php echo round(($stats['today_attendance']['presente'] / $stats['today_attendance']['total']) * 100) ?>%;>></div>
53                      </div>
54                  </div>
55              <?php endif; ?>
56          </div>
57      <?php endif; ?>
58  </div>
```

```

59     <!-- ■■ Stats: Docente ■■ -->
60     <?php if(Security::hasRole('docente') && isset($stats['my_courses'])): ?>
61     <div class="stats-row">
62         <div class="stat-card blue">
63             <div class="number"><?= $stats['my_courses'] ?></div>
64             <div class="label">Mis Cursos</div>
65         </div>
66         <div class="stat-card green">
67             <div class="number"><?= $stats['my_students'] ?></div>
68             <div class="label">Mis Estudiantes</div>
69         </div>
70         <?php if(!empty($stats['tutor_course'])): ?>
71         <div class="stat-card purple">
72             <div style="font-size:1.4rem;margin-bottom:4px;">■</div>
73             <div class="number" style="font-size:1rem;"><?= htmlspecialchars($stats['tutor_course']) ?></div>
74             <div class="label">Mi Curso como Tutor</div>
75         </div>
76         <?php endif; ?>
77     </div>
78     <?php endif; ?>
79
80     <!-- ■■ Stats: Estudiante ■■ -->
81     <?php if(Security::hasRole('estudiante') && isset($stats['my_attendance'])): ?>
82     <?php $pctAsis = $stats['my_attendance']['total'] > 0 ? round(($stats['my_attendance']['presente'] / $stats['m.
83     <div class="stats-row">
84         <div class="stat-card green">
85             <div class="number"><?= $pctAsis ?>%</div>
86             <div class="label">Mi Asistencia (30 días)</div>
87             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pctAsis ?>%;background:
88         </div>
89         <div class="stat-card red">
90             <div class="number"><?= $stats['my_attendance']['ausente'] ?></div>
91             <div class="label">Ausencias</div>
92         </div>
93         <div class="stat-card yellow">
94             <div class="number"><?= $stats['my_attendance']['tardanza'] ?></div>
95             <div class="label">Tardanzas</div>
96         </div>
97     </div>
98     <?php endif; ?>
99
100    <!-- ■■ Stats: Representante ■■ -->
101    <?php if(Security::hasRole('representante') && isset($stats['my_children'])): ?>
102    <div class="stats-row">
103        <div class="stat-card blue">
104            <div class="number"><?= $stats['my_children'] ?></div>
105            <div class="label">Representados</div>
106        </div>
107    </div>
108    <?php endif; ?>
109
110    <!-- ■■ Menú de accesos rápidos ■■ -->
111    <div class="section-title">Accesos rápidos</div>
112    <div style="display:grid;grid-template-columns:repeat(auto-fill,minmax(240px,1fr));gap:14px;margin-bottom:24px;">
113
114        <?php if(Security::hasRole(['docente','autoridad'])): ?>
115        <a href="?action=attendance_register" class="menu-card green">
116            <div class="mc-icon">■</div>
117            <div class="mc-title">Registrar Asistencia</div>

```



```

118         <div class="mc-desc">Tomar asistencia de estudiantes</div>
119     </a>
120 <?php endif; ?>
121
122 <?php if(Security::hasRole('docente')): ?>
123 <a href="?action=tutor_course_attendance" class="menu-card blue">
124     <div class="mc-icon">■</div>
125     <div class="mc-title">Asistencia de Mi Curso</div>
126     <div class="mc-desc">Ver asistencia del curso que tutoro</div>
127 </a>
128 <?php endif; ?>
129
130 <?php if(Security::hasRole(['docente','inspector','autoridad'])): ?>
131 <a href="?action=attendance_view" class="menu-card blue">
132     <div class="mc-icon">■</div>
133     <div class="mc-title">Ver Asistencias</div>
134     <div class="mc-desc">Consultar registros de asistencia</div>
135 </a>
136 <a href="?action=attendance_calendar" class="menu-card orange">
137     <div class="mc-icon">■</div>
138     <div class="mc-title">Calendario</div>
139     <div class="mc-desc">Vista mensual de asistencias</div>
140 </a>
141 <?php endif; ?>
142
143 <?php if(Security::hasRole('estudiante')): ?>
144 <a href="?action=my_attendance" class="menu-card blue">
145     <div class="mc-icon">■</div>
146     <div class="mc-title">Mi Asistencia</div>
147     <div class="mc-desc">Ver mi historial de asistencia</div>
148 </a>
149 <a href="?action=my_justifications" class="menu-card green">
150     <div class="mc-icon">■</div>
151     <div class="mc-title">Mis Justificaciones</div>
152     <div class="mc-desc">Justificar ausencias</div>
153 </a>
154 <?php endif; ?>
155
156 <?php if(Security::hasRole(['autoridad','inspector'])): ?>
157 <a href="?action=pending_justifications" class="menu-card orange">
158     <div class="mc-icon">■</div>
159     <div class="mc-title">Revisar Justificaciones</div>
160     <div class="mc-desc">Aprobar o rechazar justificaciones pendientes</div>
161 </a>
162 <?php endif; ?>
163
164 <?php if(Security::hasRole('autoridad')): ?>
165 <a href="?action=stats" class="menu-card purple">
166     <div class="mc-icon">■</div>
167     <div class="mc-title">Estadísticas</div>
168     <div class="mc-desc">Análisis y métricas de asistencia</div>
169 </a>
170 <a href="?action=reports" class="menu-card purple">
171     <div class="mc-icon">■</div>
172     <div class="mc-title">Reportes</div>
173     <div class="mc-desc">Generar informes PDF y Excel</div>
174 </a>
175 <a href="?action=users" class="menu-card blue">
176     <div class="mc-icon">■</div>

```

```

177         <div class="mc-title">Gestión de Usuarios</div>
178         <div class="mc-desc">Administrar usuarios y roles</div>
179     </a>
180     <a href="?action=academic" class="menu-card teal">
181         <div class="mc-icon">■</div>
182         <div class="mc-title">Configuración Académica</div>
183         <div class="mc-desc">Cursos, materias, matrículas</div>
184     </a>
185     <a href="?action=assignments" class="menu-card green">
186         <div class="mc-icon">■</div>
187         <div class="mc-title">Asignaciones Docentes</div>
188         <div class="mc-desc">Gestionar docentes y tutores</div>
189     </a>
190     <a href="?action=manage_representatives" class="menu-card orange">
191         <div class="mc-icon">■■■■■■■</div>
192         <div class="mc-title">Gestión Representantes</div>
193         <div class="mc-desc">Asignar representantes a estudiantes</div>
194     </a>
195     <a href="?action=backups" class="menu-card dark">
196         <div class="mc-icon">■</div>
197         <div class="mc-title">RespalDOS</div>
198         <div class="mc-desc">Gestionar respaldos de base de datos</div>
199     </a>
200     <?php endif; ?>
201
202     <?php if(Security::hasRole('representante')): ?>
203     <a href="?action=my_children" class="menu-card purple">
204         <div class="mc-icon">■■■■■■■</div>
205         <div class="mc-title">Mis Representados</div>
206         <div class="mc-desc">Ver estudiantes y su asistencia</div>
207     </a>
208     <?php endif; ?>
209
210 </div>
211 </div>
212
213 <style>
214 /* Menu cards del dashboard */
215 .menu-card {
216     display: flex;
217     flex-direction: column;
218     background: #fff;
219     border: 1px solid #e0e0e0;
220     border-left: 4px solid #007bff;
221     border-radius: 8px;
222     padding: 20px;
223     text-decoration: none;
224     color: #333;
225     transition: transform 0.2s, box-shadow 0.2s;
226     cursor: pointer;
227 }
228 .menu-card:hover {
229     transform: translateY(-3px);
230     box-shadow: 0 6px 20px rgba(0,0,0,0.12);
231     text-decoration: none;
232 }
233 .menu-card .mc-icon { font-size: 1.8rem; margin-bottom: 8px; }
234 .menu-card .mc-title { font-size: 0.95rem; font-weight: 700; margin-bottom: 4px; }
235 .menu-card .mc-desc { font-size: 0.8rem; color: #888; }

```

```
236
237 .menu-card.blue { border-left-color: #007bff; } .menu-card.blue .mc-title { color: #007bff; }
238 .menu-card.green { border-left-color: #28a745; } .menu-card.green .mc-title { color: #28a745; }
239 .menu-card.orange { border-left-color: #fd7e14; } .menu-card.orange .mc-title { color: #fd7e14; }
240 .menu-card.purple { border-left-color: #6f42c1; } .menu-card.purple .mc-title { color: #6f42c1; }
241 .menu-card.teal { border-left-color: #17a2b8; } .menu-card.teal .mc-title { color: #17a2b8; }
242 .menu-card.dark { border-left-color: #343a40; } .menu-card.dark .mc-title { color: #343a40; }
243 </style>
244
245 </body>
246 </html>
```

77. ArchImp/views/institution/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Configuración de Institución - EcuAsist</title>
8      <style>
9          /* ■■ Jornadas toggle ■■ */
10         .shifts-grid { display: flex; gap: 12px; flex-wrap: wrap; margin-top: 4px; }
11         .shift-card {
12             display: flex; align-items: center; gap: 10px;
13             padding: 12px 20px; border-radius: 10px; cursor: pointer;
14             border: 2px solid #e0e0e0; background: #f9f9f9;
15             transition: all .2s; user-select: none; min-width: 150px;
16         }
17         .shift-card:hover { border-color: #90caf9; background: #e3f2fd; }
18         .shift-card.active { border-color: #1976d2; background: #e3f2fd; }
19         .shift-card .shift-icon { font-size: 22px; }
20         .shift-card .shift-info { flex: 1; }
21         .shift-card .shift-name { font-weight: 600; font-size: 14px; color: #333; }
22         .shift-card .shift-status { font-size: 11px; margin-top: 2px; }
23         .shift-card.active .shift-status { color: #1565c0; }
24         .shift-card:not(.active) .shift-status { color: #999; }
25         .shift-card .shift-toggle {
26             width: 36px; height: 20px; border-radius: 10px; position: relative;
27             background: #ccc; transition: background .2s; flex-shrink: 0;
28         }
29         .shift-card.active .shift-toggle { background: #1976d2; }
30         .shift-card .shift-toggle::after {
31             content: ''; position: absolute; top: 2px; left: 2px;
32             width: 16px; height: 16px; border-radius: 50%; background: #fff;
33             transition: left .2s; box-shadow: 0 1px 3px rgba(0,0,0,.2);
34         }
35         .shift-card.active .shift-toggle::after { left: 18px; }
36         .shift-saving { opacity: .5; pointer-events: none; }
37
38         /* ■■ Logo preview ■■ */
39         .logo-wrap { display: flex; align-items: center; gap: 20px; flex-wrap: wrap; }
40         .logo-preview {
41             width: 100px; height: 100px; border-radius: 8px; border: 2px dashed #ddd;
42             display: flex; align-items: center; justify-content: center;
43             background: #fafafa; overflow: hidden; flex-shrink: 0;
44         }
45         .logo-preview img { max-width: 100%; max-height: 100%; object-fit: contain; }
46         .logo-preview.no-logo { font-size: 32px; color: #ccc; }
47         .logo-upload-btn {
48             display: inline-block; padding: 8px 16px; background: #f0f7ff;
49             border: 1.5px solid #90caf9; border-radius: 6px; cursor: pointer;
50             font-size: 13px; color: #1565c0; transition: all .2s;
51         }
52         .logo-upload-btn:hover { background: #e3f2fd; }
53         #logo-input { display: none; }
54     </style>
55 </head>
56 <body>
57
58 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
```

```

59
60 <div class="breadcrumb">
61     <a href="?action=dashboard">■ Inicio</a> &rsaquo; Configuración de Institución
62 </div>
63
64 <div class="container">
65
66     <?php if(isset($_GET['success'])): ?>
67         <div class="alert alert-success">✓ Información actualizada correctamente</div>
68     <?php endif; ?>
69
70     <div class="page-header" style="background:linear-gradient(135deg,#1a237e,#283593);">
71         <div class="ph-icon">■</div>
72     </div>
73     <h1>Configuración de Institución</h1>
74     <p>Datos generales, jornadas y logotipo</p>
75 </div>
76 </div>
77
78 <form method="POST" action="?action=update_institution" enctype="multipart/form-data" id="instForm">
79     <!-- Respaldo: logo actual en caso de que no se suba archivo nuevo -->
80     <input type="hidden" name="current_logo_path" value="<?php htmlspecialchars($institution['logo_path']) ?>" />
81
82     <div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">
83
84         <!-- ■■ Panel izquierdo: datos ■■ -->
85     </div>
86     <div class="panel" style="margin-bottom:16px;">
87         <h3 style="font-size:.95rem;color:#555;margin-bottom:14px;">■ Datos de la Institución</h3>
88         <div class="form-group">
89             <label>Nombre de la institución *</label>
90             <input type="text" name="name" class="form-control" required
91                 value="<?php htmlspecialchars($institution['name']) ?>" />
92         </div>
93         <div class="form-row">
94             <div class="form-group">
95                 <label>Código AMIE</label>
96                 <input type="text" name="amie_code" class="form-control"
97                     value="<?php htmlspecialchars($institution['amie_code']) ?>"
98                     placeholder="17H01988" />
99             </div>
100             <div class="form-group">
101                 <label>Teléfono</label>
102                 <input type="text" name="phone" class="form-control"
103                     value="<?php htmlspecialchars($institution['phone']) ?>"
104                     placeholder="02-2345678" />
105             </div>
106         </div>
107         <div class="form-group">
108             <label>Email institucional</label>
109             <input type="email" name="email" class="form-control"
110                 value="<?php htmlspecialchars($institution['email']) ?>"
111                 placeholder="info@institucion.edu.ec" />
112         </div>
113         <div class="form-group">
114             <label>Sitio web</label>
115             <input type="text" name="website" class="form-control" id="website"
116                 value="<?php htmlspecialchars($institution['website']) ?>"
117                 placeholder="www.institucion.edu.ec" />

```

```

118         onblur="autoHttps(this)"&gt;
119     </div>&gt;
120     <div class="form-group"&gt;
121         <label>Nombre del Director/Rector</label>
122         <input type="text" name="director_name" class="form-control"
123             value="&lt;?= htmlspecialchars($institution['director_name']) ?&gt;"
124             placeholder="MSc. Nombre Apellido"&gt;
125     </div>&gt;
126 </div>&gt;
127
128 <div class="panel" style="margin-bottom:16px;"&gt;
129     <h3 style="font-size:.95rem;color:#555;margin-bottom:14px;"&gt;■ Ubicación</h3>
130     <div class="form-row"&gt;
131         <div class="form-group"&gt;
132             <label>Provincia</label>
133             <select name="province" id="province" class="form-control" onchange="loadCities()"&gt;
134                 <option value=""&gt;Seleccionar...</option>
135                 <?php foreach(getProvinces() as $prov): ?&gt;
136                     <option value="&lt;?= $prov ?&gt;" &lt;?= ($institution['province']) ?&gt; == $prov ?&gt;
137                         &lt;?= $prov ?&gt;
138                 </option>
139                 <?php endforeach; ?&gt;
140             </select>
141         </div>
142         <div class="form-group"&gt;
143             <label>Ciudad</label>
144             <select name="city" id="city" class="form-control"&gt;
145                 <option value="&lt;?= htmlspecialchars($institution['city']) ?&gt;"&gt;
146                     &lt;?= htmlspecialchars($institution['city']) ?&gt; 'Seleccionar provincia...' ?&gt;
147                 </option>
148             </select>
149         </div>
150     </div>
151     <div class="form-group"&gt;
152         <label>Dirección</label>
153         <input type="text" name="address" class="form-control"
154             value="&lt;?= htmlspecialchars($institution['address']) ?&gt;"
155             placeholder="Av. Principal 123"&gt;
156     </div>
157 </div>
158 </div>
159
160 <!-- ■■ Panel derecho: logo + jornadas ■■ --&gt;
161 <div>
162     <!-- Logo --&gt;
163     <div class="panel" style="margin-bottom:16px;"&gt;
164         <h3 style="font-size:.95rem;color:#555;margin-bottom:14px;"&gt;■■ Logo Institucional</h3>
165         <div class="logo-wrap"&gt;
166             <div class="logo-preview" id="logoPreview"&gt;
167                 <?php
168                 // BASE_URL viene de env.php (local o producción)
169                 $logoUrl = '';
170                 if (!empty($institution['logo_path'])) {
171                     $imgFile = ltrim(str_replace('uploads/', '', $institution['logo_path']), '/');
172                     $logoUrl = BASE_URL . '/img.php?f=' . urlencode($imgFile) . '&amp;v=' . time();
173                 }
174                 ?&gt;
175                 <?php if($logoUrl): ?&gt;
176                     
178             <span class="no-logo" id="noLogoIcon">■</span>
179         <?php endif: ?>
180     </div>
181 </div>
182     <label for="logo-input" class="logo-upload-btn">■ Seleccionar imagen</label>
183     <input type="file" name="logo" id="logo-input" accept=".jpg,.jpeg,.png,.gif,.webp"
184         onchange="previewLogo(this)">
185     <p style="font-size:12px;color:#999;margin-top:6px;">JPG, PNG o WebP – máx. 2MB</p>
186     <?php if($logoUrl): ?>
187     <p style="font-size:12px;color:#2e7d32;margin-top:4px;">✓ Logo actual cargado</p>
188     <?php endif: ?>
189 </div>
190 </div>
191 </div>
192
193 <!-- Jornadas toggle -->
194 <div class="panel">
195     <h3 style="font-size:.95rem;color:#555;margin-bottom:6px;">■ Jornadas de la Institución</h3>
196     <p style="font-size:12px;color:#999;margin-bottom:14px;">Clic para activar o desactivar cada j.
197
198     <?php
199     $shiftIcons = [
200         'matutina' => [ '■', 'Mañana'],
201         'vespertina' => [ '■', 'Tarde'],
202         'nocturna' => [ '■', 'Noche'],
203     ];
204     ?>
205     <div class="shifts-grid" id="shiftsGrid">
206         <?php foreach($allShifts as $shift):
207             $isActive = in_array($shift['id'], $assignedShiftIds);
208             $icon      = $shiftIcons[strtolower($shift['name'])][0] ?? '■';
209             $timeLabel= $shiftIcons[strtolower($shift['name'])][1] ?? ucfirst($shift['name']);
210             ?>
211             <div class="shift-card <?php $isActive ? 'active' : '' ?>"
212                 id="shift-<?php $shift['id'] ?>"
213                 onclick="toggleShift(<?php $shift['id'] ?>, this)"
214                 title="Clic para <?php $isActive ? 'desactivar' : 'activar' ?>">
215                 <span class="shift-icon"><?php $icon ?></span>
216                 <div class="shift-info">
217                     <div class="shift-name"><?php ucfirst($shift['name']) ?></div>
218                     <div class="shift-status"><?php $isActive ? '✓ Activa' : 'Inactiva' ?></div>
219                 </div>
220                 <div class="shift-toggle"></div>
221             </div>
222             <?php endforeach: ?>
223         </div>
224
225         <div id="shift-msg" style="font-size:12px;margin-top:10px;min-height:18px;"></div>
226     </div>
227 </div>
228 </div>
229
230 <div style="display:flex;gap:10px;">
231     <button type="submit" class="btn btn-success">■ Guardar Cambios</button>
232     <a href="?action=dashboard" class="btn btn-outline">Cancelar</a>
233 </div>
234
235 </form>

```

[illegible]

[illegible]

```
354         }
355     }
356 }
357 }());
358 </script>
359
360 <?php
361 function getProvinces() {
362     return ['Azuary', 'Bolívar', 'Cañar', 'Carchi', 'Chimborazo', 'Cotopaxi', 'El Oro',
363         'Esmeraldas', 'Galápagos', 'Guayas', 'Imbabura', 'Loja', 'Los Ríos', 'Manabí',
364         'Morona Santiago', 'Napo', 'Orellana', 'Pastaza', 'Pichincha', 'Santa Elena',
365         'Santo Domingo', 'Sucumbíos', 'Tungurahua', 'Zamora Chinchipe'];
366 }
367 ?>
368
369 </body>
370 </html>
```

78. ArchImp/views/justifications/my_list.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mis Justificaciones - EcuAsist</title>
8      <style>
9          .badge-pending { background:#ffc107; color:#000; padding:5px 12px; border-radius:12px; font-size:12px; font-w.
10         .badge-approved { background:#28a745; color:white; padding:5px 12px; border-radius:12px; font-size:12px; font-w.
11         .badge-rejected { background:#dc3545; color:white; padding:5px 12px; border-radius:12px; font-size:12px; font-w.
12         .stat-box { background:white; border-radius:8px; padding:20px; text-align:center;
13             box-shadow:0 2px 8px rgba(0,0,0,.08); flex:1; }
14         .stat-box .num { font-size:28px; font-weight:700; }
15         .stat-box .lbl { font-size:12px; color:#888; margin-top:4px; }
16     </style>
17 </head>
18 <body>
19
20 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
21
22 <div class="breadcrumb">
23     <a href="?action=dashboard">■ Inicio</a> &rsquo; Mis Justificaciones
24 </div>
25
26 <div class="container">
27
28     <?php if(isset($_GET['success'])): ?>
29         <div class="alert alert-success">✓ Justificación enviada correctamente. Está pendiente de revisión.</di.
30     <?php endif; ?>
31
32     <div class="page-header" style="background:linear-gradient(135deg,#e65100,#ef6c00);display:flex;justify-content:s.
33         <div style="display:flex;align-items:center;gap:12px;">
34             <div class="ph-icon">■</div>
35             <div>
36                 <h1>Mis Justificaciones</h1>
37                 <p>Historial de justificaciones enviadas</p>
38             </div>
39         </div>
40         <a href="?action=submit_justification"
41             style="padding:9px 18px;background:white;color:#e65100;text-decoration:none;
42             border-radius:6px;font-weight:600;font-size:13px;white-space:nowrap;">
43             ■ Nueva Justificación
44         </a>
45     </div>
46
47     <?php if(empty($justifications)): ?>
48         <div class="empty-state">
49             <div class="icon">■</div>
50             <p>No tienes justificaciones enviadas aún.</p>
51             <p style="font-size:13px;color:#aaa;margin-top:6px;">Puedes justificar ausencias desde "Mi Asistencia".
52             <a href="?action=my_attendance" class="btn btn-primary" style="margin-top:16px;">← Ir a Mi Asistencia&
53         </div>
54
55     <?php else: ?>
56
57         <!-- Resumen -->
58         <div style="display:flex;gap:16px;margin-bottom:20px;flex-wrap:wrap;">
```

```

59         <div class="stat-box">
60             <div class="num" style="color:#f57c00;">
61                 <?= count(array_filter($justifications, fn($j) => $j['status'] === 'pendiente')) ?>
62             </div>
63             <div class="lbl">■ Pendientes</div>
64         </div>
65         <div class="stat-box">
66             <div class="num" style="color:#2e7d32;">
67                 <?= count(array_filter($justifications, fn($j) => $j['status'] === 'aprobado')) ?>
68             </div>
69             <div class="lbl">✓ Aprobadas</div>
70         </div>
71         <div class="stat-box">
72             <div class="num" style="color:#c62828;">
73                 <?= count(array_filter($justifications, fn($j) => $j['status'] === 'rechazado')) ?>
74             </div>
75             <div class="lbl">✗ Rechazadas</div>
76         </div>
77     </div>
78
79     <!-- Tabla -->
80     <div class="panel">
81         <table>
82             <thead>
83                 <tr>
84                     <th>#</th>
85                     <th>Fecha</th>
86                     <th>Motivo</th>
87                     <th>Documento</th>
88                     <th>Estado</th>
89                     <th>Envío</th>
90                     <th>Revisado por</th>
91                     <th>Observaciones</th>
92                 </tr>
93             </thead>
94             <tbody>
95                 <?php $i = 1; foreach($justifications as $just): ?>
96                     <tr>
97                         <td><?= $i++ ?></td>
98                         <td><?= date('d/m/Y', strtotime($just['attendance_date'])) ?></td>
99                         <td><?= htmlspecialchars($just['reason']) ?></td>
100                        <td>
101                            <?php if($just['document_path']): ?>
102                                <a href="<?= BASE_URL ?>/<?= ltrim($just['document_path'],'/') ?>"
103                                    target="_blank"
104                                    style="padding:5px 10px;background:#007bff;color:white;border-radius:4px;
105                                        text-decoration:none;font-size:12px;display:inline-block;">
106                                    ■ Ver
107                                </a>
108                            <?php else: ?>
109                                <span style="color:#aaa;font-size:12px;">Sin documento</span>
110                            <?php endif; ?>
111                        </td>
112                        <td>
113                            <?php if($just['status'] === 'pendiente'): ?>
114                                <span class="badge-pending">■ Pendiente</span>
115                            <?php elseif($just['status'] === 'aprobado'): ?>
116                                <span class="badge-approved">✓ Aprobado</span>
117                            <?php else: ?>

```

```

118         <span class="badge-rejected">✖ Rechazado</span>
119     <?php endif; ?>
120 </td>
121 <td style="font-size:12px;"><?= date('d/m/Y H:i', strtotime($just['submitted_at'])) ?&
122 <td style="font-size:12px;">
123     <?php if($just['reviewed_by']): ?>
124         <?= htmlspecialchars($just['reviewer_name']) ?><br>
125         <span style="color:#888;"><?= date('d/m/Y H:i', strtotime($just['reviewed_at'])).
126     <?php else: ?>
127         <span style="color:#aaa;">—</span>
128     <?php endif; ?>
129 </td>
130 <td style="font-size:12px;">
131     <?= $just['review_notes'] ? htmlspecialchars($just['review_notes']) : '
132 </td>
133 </tr>
134 <?php endforeach; ?>
135 </tbody>
136 </table>
137 </div>
138
139 <div style="margin-top:16px;">
140     <a href="?action=my_attendance" class="btn btn-outline">← Volver a Mi Asistencia</a>
141 </div>
142
143 <?php endif; ?>
144
145 </div>
146 </body>
147 </html>

```

79. ArchImp/views/justifications/pending.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Justificaciones Pendientes - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo; Justificaciones Pendientes
15  </div>
16
17  <div class="container">
18
19      <?php if(isset($_GET['success'])): ?>
20          <div class="alert alert-success">✓ Justificación procesada correctamente</div>
21      <?php endif; ?>
22
23      <div class="page-header" style="background:linear-gradient(135deg,#1b5e20,#2e7d32);">
24          <div class="ph-icon">■</div>
25      </div>
26      <h1>Justificaciones Pendientes</h1>
27      <p><?= count($justifications) ?> solicitud<?= count($justifications)!=1?'es':'' ?> por rev.
28  </div>
29  </div>
30
31  <?php if(empty($justifications)): ?>
32      <div class="empty-state">
33          <div class="icon">■</div>
34          <p>No hay justificaciones pendientes de revisión.</p>
35      </div>
36
37  <?php else: ?>
38      <div class="table-wrap">
39          <table>
40              <thead>
41                  <tr>
42                      <th>#</th>
43                      <th>Estudiante</th>
44                      <th>Curso</th>
45                      <th>Período de ausencia</th>
46                      <th style="text-align:center;">Días Lab.</th>
47                      <th style="text-align:center;">Revisión</th>
48                      <th>Enviado por</th>
49                      <th>Fecha solicitud</th>
50                      <th>Acciones</th>
51                  </tr>
52              </thead>
53              <tbody>
54                  <?php $i = 1; foreach($justifications as $just):
55                      $days = (int)($just['working_days'] ?? 1);
56                      $canApprove = $just['can_approve'] ?? 'inspector';
57
58                      if ($canApprove === 'tutor') {
```

```

59         $approverLabel = '■ Tutor';
60         $approverColor = 'badge-green';
61     } elseif ($canApprove === 'inspector') {
62         $approverLabel = '■ Inspector';
63         $approverColor = 'badge-yellow';
64     } else {
65         $approverLabel = '■ Autoridad';
66         $approverColor = 'badge-red';
67     }
68
69     // Período de ausencia
70     if (!empty($just['date_from'])) {
71         $dateFrom = date('d/m/Y', strtotime($just['date_from']));
72         $dateTo = !empty($just['date_to']) && $just['date_to'] !== $just['date_from']
73             ? date('d/m/Y', strtotime($just['date_to']))
74             : null;
75         $period = $dateTo ? "$dateFrom al $dateTo" : $dateFrom;
76     } elseif (!empty($just['attendance_date'])) {
77         $period = date('d/m/Y', strtotime($just['attendance_date']));
78     } else {
79         $period = '-';
80     }
81     ?>
82     <tr>
83         <td style="color:#999;font-size:12px;">&lt;?=$i++ ?>&lt;/td>
84         <td>&lt;strong>&lt;?=$htmlspecialchars($just['student_name']) ?>&lt;/strong>&lt;/td>&
85         <td style="font-size:13px;">&lt;?=$htmlspecialchars($just['course_name']) ?> '-' ?>&lt;/td>&
86         <td style="font-size:13px;">&lt;?=$period ?>&lt;/td>
87         <td style="text-align:center;">
88             &lt;span style="font-weight:700;color:&lt;?=$days&lt;?=3?'#2e7d32':'#f57f17' ?>">
89                 &lt;?=$days ?>
90             &lt;/span>
91         </td>
92         <td style="text-align:center;">
93             &lt;span class="badge &lt;?=$approverColor ?>">&lt;?=$approverLabel ?>&lt;/span>
94         </td>
95         <td style="font-size:13px;color:#666;">&lt;?=$htmlspecialchars($just['submitted_by_name']) ?&
96         <td style="font-size:12px;color:#999;">&lt;?=$date('d/m/Y H:i', strtotime($just['created_at']))&
97         </td>
98         <button class="btn btn-primary btn-sm"
99             onclick="openReview(
100                 &lt;?=$just['id'] ?>,
101                 &lt;?=$json_encode($just['reason']) ?>,
102                 &lt;?=$json_encode($just['document_path']) ?>,
103                 &lt;?=$json_encode($period) ?>,
104                 &lt;?=$days ?>
105             )">■ Revisar&lt;/button>
106     </td>
107 </tr>
108 <?php endforeach; ?>
109 </tbody>
110 </table>
111 </div>
112 <?php endif; ?>
113
114 </div>
115
116 <!-- Modal revisar -->
117 <div class="modal-overlay" id="modalReview">

```

```

118     <div class="modal-box" style="max-width:520px;">
119         <h3 style="margin-bottom:16px;">■ Revisar Justificación</h3>
120
121         <div style="background:#f8f9fa;border-radius:6px;padding:12px;margin-bottom:14px;font-size:13px;">
122             <div><strong>Período:</strong> <span id="rv-period"></span></div>
123             <div style="margin-top:4px;"><strong>Días laborables:</strong> <span id="rv-days"></span></div>
124         </div>
125
126         <div style="margin-bottom:12px;">
127             <strong style="font-size:13px;">Motivo:</strong>
128             <p id="rv-reason" style="margin-top:6px;font-size:14px;color:#444;line-height:1.5;"></p>
129         </div>
130
131         <div id="rv-doc" style="margin-bottom:14px;display:none;">
132             <strong style="font-size:13px;">Documento adjunto:</strong>
133             <div style="margin-top:6px;">
134                 <a id="rv-doc-link" href="#" target="_blank" class="btn btn-info btn-sm">■ Ver documento</a>
135             </div>
136         </div>
137
138         <form method="POST" action="?action=review_justification">
139             <input type="hidden" name="justification_id" id="rv-id">
140             <div class="form-group">
141                 <label style="font-size:13px;">Observaciones <span style="color:#999;font-weight:400;">(opcional)</span>
142                 <textarea name="notes" class="form-control" rows="3" placeholder="Notas para el estudiante...">
143             </div>
144             <div class="modal-actions">
145                 <button type="button" class="btn btn-outline" onclick="closeModal('modalReview')">Cancelar</button>
146                 <button type="submit" name="review_action" value="reject" class="btn btn-danger">✗ Rechazar</button>
147                 <button type="submit" name="review_action" value="approve" class="btn btn-success">✓ Aprobar</button>
148             </div>
149         </form>
150     </div>
151 </div>
152
153 <script>
154 function openReview(id, reason, docPath, period, days) {
155     document.getElementById('rv-id').value = id;
156     document.getElementById('rv-period').textContent = period;
157     document.getElementById('rv-days').textContent = days + ' día' + (days !== 1 ? 's' : '');
158     document.getElementById('rv-reason').textContent = reason || '(sin descripción)';
159
160     var docBlock = document.getElementById('rv-doc');
161     if (docPath) {
162         document.getElementById('rv-doc-link').href = '&lt;?=' + BASE_URL + '&gt;' + docPath;
163         docBlock.style.display = 'block';
164     } else {
165         docBlock.style.display = 'none';
166     }
167     document.getElementById('modalReview').classList.add('on');
168 }
169 function openModal(id) { document.getElementById(id).classList.add('on'); }
170 function closeModal(id) { document.getElementById(id).classList.remove('on'); }
171 document.querySelectorAll('.modal-overlay').forEach(function(m) {
172     m.addEventListener('click', function(e){ if(e.target === m) closeModal(m.id); });
173 });
174 </script>
175
176 </body>

```


80. ArchImp/views/justifications/reviewed.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Justificaciones Revisadas - EcuAsist</title>
8      <style>
9          .badge-aprobado { background:#d4edda; color:#155724; padding:4px 10px; border-radius:12px; font-size:12px; font-
10         .badge-rechazado{ background:#f8d7da; color:#721c24; padding:4px 10px; border-radius:12px; font-size:12px; font-
11         .filter-bar { display:flex; gap:8px; flex-wrap:wrap; margin-bottom:20px; }
12         .filter-bar a { padding:6px 16px; border-radius:20px; text-decoration:none; font-size:13px;
13             font-weight:600; border:2px solid; transition:all .15s; }
14         .modal { display:none; position:fixed; top:0; left:0; width:100%; height:100%;
15             background:rgba(0,0,0,.5); z-index:1000; align-items:center; justify-content:center; }
16         .modal.open { display:flex; }
17         .modal-content { background:white; padding:30px; border-radius:8px; max-width:560px;
18             width:90%; box-shadow:0 4px 20px rgba(0,0,0,.3); position:relative; }
19         .modal-close { position:absolute; top:12px; right:16px; font-size:24px;
20             cursor:pointer; color:#888; line-height:1; }
21         .modal-close:hover { color:#333; }
22     </style>
23 </head>
24 <body>
25
26 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
27
28 <div class="breadcrumb">
29     <a href="?action=dashboard">■ Inicio</a> &rsquo; Justificaciones Revisadas
30 </div>
31
32 <div class="container">
33
34     <?php if(isset($_GET['success'])): ?>
35         <div class="alert alert-success">✓ Acción realizada correctamente</div>
36     <?php endif; ?>
37
38     <div class="page-header" style="background:linear-gradient(135deg,#1b5e20,#388e3c);display:flex;justify-content:s
39         <div style="display:flex;align-items:center;gap:12px;">
40             <div class="ph-icon">■</div>
41             <div>
42                 <h1>Justificaciones Revisadas</h1>
43                 <p>Historial de justificaciones aprobadas y rechazadas</p>
44             </div>
45         </div>
46         <a href="?action=pending_justifications"
47             style="padding:9px 18px;background:white;color:#1b5e20;text-decoration:none;
48                 border-radius:6px;font-weight:600;font-size:13px;white-space:nowrap;">
49             ■ Ver Pendientes
50         </a>
51     </div>
52
53     <!-- Filtros -->
54     <div class="filter-bar">
55         <a href="?action=reviewed_justifications"
56             style="<?=( $filter==='all') ? 'background:#343a40;color:white;border-color:#343a40;' : 'background:white;
57             ■ Todas (<?=( count($justifications) ?>
58         </a>
```

```

59         <a href="?action=reviewed_justifications&filter=aprobado"
60         style="&lt;?=( $filter==='aprobado') ? 'background:#28a745;color:white;border-color:#28a745;' : 'background:w
61         ■ Aprobadas
62     </a>
63     <a href="?action=reviewed_justifications&filter=rechazado"
64     style="&lt;?=( $filter==='rechazado') ? 'background:#dc3545;color:white;border-color:#dc3545;' : 'background:
65     ■ Rechazadas
66 </a>
67 </div>
68
69 <?php if(empty($justifications)): ?>
70     <div class="empty-state">
71         <div class="icon">■</div>
72         <p>No hay justificaciones revisadas con este filtro.</p>
73     </div>
74 <?php else: ?>
75 <div class="panel">
76     <table>
77         <thead>
78             <tr>
79                 <th>#</th>
80                 <th>Estudiante</th>
81                 <th>Curso</th>
82                 <th>Asignatura</th>
83                 <th>Fecha Falta</th>
84                 <th>Enviado por</th>
85                 <th>Estado</th>
86                 <th>Revisado por</th>
87                 <th>Acciones</th>
88             </tr>
89         </thead>
90         <tbody>
91             <?php $i = 1; foreach($justifications as $j):
92                 $jData = htmlspecialchars(json_encode([
93                     'student' => $j['student_name'],
94                     'reason' => $j['reason'],
95                     'notes' => $j['review_notes'] ?? '',
96                     'doc' => $j['document_path'] ?? '',
97                     'status' => $j['status'],
98                 ]), ENT_QUOTES);
99             ?>
100             <tr>
101                 <td>&lt;?=$i++ ?>&lt;/td>
102                 <td>&lt;?=$j['student_name'] ?>&lt;/td>
103                 <td>&lt;?=$j['course_name'] ?? '-' ?>&lt;/td>
104                 <td>&lt;?=$j['subject_name'] ?>&lt;/td>
105                 <td>&lt;?= date('d/m/Y', strtotime($j['attendance_date'])) ?>&lt;/td>
106                 <td>&lt;?=$j['submitted_by_name'] ?>&lt;/td>
107                 <td>
108                     <?php if($j['status'] === 'aprobado'): ?>
109                         <span class="badge-aprobado">■ Aprobada</span>
110                     <?php else: ?>
111                         <span class="badge-rechazado">■ Rechazada</span>
112                     <?php endif: ?>
113                 </td>
114                 <td>&lt;?=$j['reviewer_name'] ?? '-' ?>&lt;/td>
115                 <td>
116                     <button class="btn btn-primary"
117                     style="padding:5px 12px;font-size:12px;"

```

```

118                 onclick="verDetalle('&lt;?= $jData ?&gt;')"&gt;■ Ver&lt;/button&gt;
119             &lt;/td&gt;
120         &lt;/tr&gt;
121     &lt;?php endforeach; ?&gt;
122     &lt;/tbody&gt;
123 &lt;/table&gt;
124 &lt;/div&gt;
125 &lt;?php endif; ?&gt;
126
127 &lt;/div&gt;
128
129 &lt;!-- Modal detalle --&gt;
130 &lt;div id="modal" class="modal" onclick="if(event.target===this)cerrarModal()"&gt;
131     &lt;div class="modal-content"&gt;
132         &lt;span class="modal-close" onclick="cerrarModal()"&gt;&amp;times;&lt;/span&gt;
133         &lt;h3 style="margin-bottom:16px;"&gt;■ Detalle de Justificación&lt;/h3&gt;
134         &lt;div id="modal-body"&gt;&lt;/div&gt;
135     &lt;/div&gt;
136 &lt;/div&gt;
137
138 &lt;script&gt;
139 function verDetalle(jsonStr) {
140     const d = JSON.parse(jsonStr);
141     const badge = d.status === 'aprobado'
142         ? '&lt;span style="background:#d4edda;color:#155724;padding:3px 10px;border-radius:10px;"&gt;■ Aprobada&lt;/span.'
143         : '&lt;span style="background:#f8d7da;color:#721c24;padding:3px 10px;border-radius:10px;"&gt;■ Rechazada&lt;/spa.
144
145     let html = `&lt;p&gt;&lt;strong&gt;Estudiante:&lt;/strong&gt; ${d.student}&lt;/p&gt;`;
146     html += `&lt;p style="margin-top:10px;"&gt;&lt;strong&gt;Estado:&lt;/strong&gt; ${badge}&lt;/p&gt;`;
147     html += `&lt;p style="margin-top:10px;"&gt;&lt;strong&gt;Motivo:&lt;/strong&gt;&lt;/p&gt;&lt;p style="margin-top:4px.
148
149     if (d.notes) {
150         html += `&lt;p style="margin-top:10px;"&gt;&lt;strong&gt;Observación del revisor:&lt;/strong&gt;&lt;/p&gt;&lt;p .
151     }
152
153     if (d.doc) {
154         html += `&lt;p style="margin-top:14px;"&gt;&lt;strong&gt;Documento:&lt;/strong&gt;
155             &lt;a href="&lt;?= BASE_URL ?&gt;/${d.doc}" target="_blank"
156                 style="color:#007bff;"&gt;■ Ver archivo adjunto&lt;/a&gt;&lt;/p&gt;`;
157     } else {
158         html += `&lt;p style="margin-top:14px;color:#999;"&gt;&lt;em&gt;Sin documento adjunto&lt;/em&gt;&lt;/p&gt;`;
159     }
160
161     document.getElementById('modal-body').innerHTML = html;
162     document.getElementById('modal').classList.add('open');
163 }
164
165 function cerrarModal() {
166     document.getElementById('modal').classList.remove('open');
167 }
168 &lt;/script&gt;
169
170 &lt;/body&gt;
171 &lt;/html&gt;

```

81. ArchImp/views/justifications/submit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Justificar Ausencia - EcuAsist</title>
8      <style>
9          .reason-grid { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; margin-bottom: 12px; }
10         .reason-option { display: flex; align-items: center; gap: 8px; padding: 10px 14px;
11             border: 2px solid #e0e0e0; border-radius: 8px; cursor: pointer;
12             transition: all .2s; font-size: 14px; background: #fff; }
13         .reason-option:hover { border-color: #007bff; background: #f0f7ff; }
14         .reason-option input[type=radio] { accent-color: #007bff; width: 16px; height: 16px; flex-shrink: 0; }
15         .reason-option.selected { border-color: #007bff; background: #e8f2ff; font-weight: 600; }
16
17         .absence-item { display: flex; align-items: center; gap: 12px; padding: 12px 16px;
18             border: 2px solid #e0e0e0; border-radius: 8px; cursor: pointer;
19             transition: all .2s; margin-bottom: 8px; background: #fff; }
20         .absence-item:hover { border-color: #007bff; background: #f0f7ff; }
21         .absence-item.checked { border-color: #007bff; background: #e8f2ff; }
22         .absence-item input[type=checkbox] { accent-color: #007bff; width: 18px; height: 18px; flex-shrink: 0; }
23         .absence-date { font-weight: 700; font-size: 15px; color: #333; min-width: 95px; }
24         .absence-meta { font-size: 13px; color: #666; flex: 1; }
25         .absence-hour { display: inline-block; font-size: 12px; background: #f0f0f0;
26             padding: 2px 8px; border-radius: 4px; color: #555; margin: 2px 2px 0 0; }
27         .select-all-bar { display: flex; align-items: center; gap: 10px;
28             padding: 8px 0; margin-bottom: 10px; border-bottom: 1px solid #eee; }
29         #otro-block, #detail-block { display: none; margin-top: 6px; }
30         .counter-badge { display: inline-block; background: #007bff; color: #fff;
31             border-radius: 12px; padding: 3px 12px; font-size: 13px; font-weight: 700; }
32         .approver-note { padding: 10px 14px; border-radius: 6px; font-size: 13px;
33             margin-top: 10px; display: none; }
34         .step-title { font-size: .95rem; color: #555; margin-bottom: 14px; font-weight: 600; }
35     </style>
36 </head>
37 <body>
38
39 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
40
41 <div class="breadcrumb">
42     <a href="?action=dashboard">■ Inicio</a> &rsquo;
43     <a href="?action=my_attendance">Mi Asistencia</a> &rsquo;
44     Justificar Ausencia
45 </div>
46
47 <div class="container" style="max-width:720px;">
48
49     <?php if(isset($_GET['error'])): ?>
50         <div class="alert alert-danger">
51             <?=[ 'no_absences' => 'Debes seleccionar al menos una ausencia.' ][$_GET['error']] ?? 'Error al enviar...'
52         </div>
53     <?php endif; ?>
54
55     <div class="page-header" style="background: linear-gradient(135deg, #e65100, #ef6c00);">
56         <div class="ph-icon">■</div>
57         <div>
58             <h1>Justificar Ausencia</h1>
```

```

59         <p>Selecciona los días que deseas justificar</p>
60     </div>
61 </div>
62
63 <?php if(empty($absences)): ?>
64 <div class="empty-state">
65     <div class="icon"><img alt="Empty state icon" /></div>
66     <p>No tienes ausencias pendientes de justificar.</p>
67     <a href="?action=my_attendance" class="btn btn-primary" style="margin-top:12px;">Volver</a>
68 </div>
69
70 <?php else: ?>
71
72 <div class="alert alert-info">
73     <strong>Selecciona uno o varios días. <strong>Hasta 3 días</strong> → revisa el <strong>Docente Tutor</strong>
74 </div>
75
76 <form method="POST" enctype="multipart/form-data" id="justForm">
77
78     <?php if(Security::hasRole('representante')): ?>
79     <input type="hidden" name="student_id" value="<?php isset($_GET['student_id']) ? (int)$_GET['student_id'] : 0;" />
80     <?php endif; ?>
81
82     <!-- PASO 1: Seleccionar días -->
83     <div class="panel" style="margin-bottom:16px;">
84         <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px;">
85             <p class="step-title" style="margin:0;">Paso 1 – Días de ausencia a justificar</p>
86             <span id="selected-count" class="counter-badge">0 seleccionados</span>
87         </div>
88
89         <div class="select-all-bar">
90             <label style="display:flex;align-items:center;gap:8px;cursor:pointer;font-size:13px;color:#555;">
91                 <input type="checkbox" id="select-all" onchange="toggleAll(this)"
92                     style="width:16px;height:16px;accent-color:#007bff;" />
93                 Seleccionar todos (<?php count($absences) > días)
94             </label>
95         </div>
96
97         <?php
98         // Agrupar registros por fecha (un día puede tener varias horas)
99         $byDate = [];
100         foreach ($absences as $a) {
101             $byDate[$a['date']][] = $a;
102         }
103         $days = ['Domingo','Lunes','Martes','Miércoles','Jueves','Viernes','Sábado'];
104         foreach ($byDate as $date => $rows):
105             $ids = implode(',', array_column($rows, 'id'));
106             $dayLabel = $days[date('w', strtotime($date))];
107             $dateLabel= date('d/m/Y', strtotime($date));
108             <?php
109             <div class="absence-item" onclick="toggleCheck(this)">
110                 <input type="checkbox" name="attendance_ids[]" value="<?php $ids >"
111                     onchange="updateCount(); event.stopPropagation();" />
112                 <div class="absence-date"><?php $dateLabel ></div>
113                 <div class="absence-meta">
114                     <div style="font-weight:600;color:#444;margin-bottom:4px;"><?php $dayLabel ></div>
115                     <?php foreach($rows as $r): ?>
116                         <span class="absence-hour">Hora <?php $r['hour_period'] > · <?php htmlspecialchars($r['description']) >
117                     <?php endforeach; ?>

```

```

118         </div>
119     </div>
120     <?php endforeach; ?>
121
122     <!-- Aviso quién aprueba -->
123     <div class="approver-note" id="note-tutor" style="background:#e8f5e9;color:#2e7d32;">
124         ■ Con <strong id="lbl-t"></strong> día(s), revisará el <strong>Docente Tutor</stro
125     </div>
126     <div class="approver-note" id="note-insp" style="background:#fff8e1;color:#f57f17;">
127         ■■ Con <strong id="lbl-i"></strong> día(s), revisará el <strong>Inspector o Autoridad
128     </div>
129 </div>
130
131 <!-- ■■ PASO 2: Motivo ■■ -->
132 <div class="panel" style="margin-bottom:16px;">
133     <p class="step-title">■ Paso 2 – Motivo de la ausencia</p>
134     <div class="reason-grid">
135         <?php foreach([
136             ['■','Enfermedad o malestar'],
137             ['■','Cita médica o examen'],
138             ['■■■■■','Fallecimiento familiar'],
139             ['■■■','Trámites legales / judiciales'],
140             ['■','Accidente o emergencia'],
141             ['■■■','Causas climáticas'],
142             ['■','Evento deportivo o académico oficial'],
143             ['✎■','Viaje o traslado familiar'],
144             ['■','Otro motivo'],
145         ] as [$icon, $label]):
146             $val = ($label === 'Otro motivo') ? 'Otro' : $label;
147         <?>
148         <label class="reason-option" onclick="selectReason(this,'<?>= addslashes($val) ?>')">
149             <input type="radio" name="reason_type" value="<?>= htmlspecialchars($val) ?>" required>
150             <span style="font-size:18px;"><?>= $icon ?></span>
151             <?>= htmlspecialchars($label) ?>
152         </label>
153     <?php endforeach; ?>
154 </div>
155 <div id="otro-block">
156     <textarea name="reason" id="reason-text" class="form-control" rows="3"
157         placeholder="Describe el motivo de la ausencia..."></textarea>
158 </div>
159 <div id="detail-block">
160     <input type="text" name="reason" id="reason-detail" class="form-control"
161         placeholder="Detalle adicional (opcional)...">
162 </div>
163 </div>
164
165 <!-- ■■ PASO 3: Documento ■■ -->
166 <div class="panel" style="margin-bottom:20px;">
167     <p class="step-title">■ Paso 3 – Documento de respaldo <span style="color:#999;font-weight:400;"&
168     <input type="file" name="document" class="form-control" accept=".pdf,.jpg,.jpeg,.png">
169     <small style="color:#888;display:block;margin-top:4px;">PDF, JPG o PNG – máximo 5MB</small>
170 </div>
171
172 <div style="display:flex;gap:10px;">
173     <button type="submit" class="btn btn-success" id="submitBtn" disabled>■ Enviar Justificación</butto
174     <a href="?action=my_attendance" class="btn btn-outline">Cancelar</a>
175 </div>
176

```

```

177     </form>;
178     <?php endif; ?>;
179
180 </div>;
181
182 </script>;
183 function toggleCheck(item) {
184     var cb = item.querySelector('input[type=checkbox]');
185     cb.checked = !cb.checked;
186     item.classList.toggle('checked', cb.checked);
187     updateCount();
188 }
189
190 function toggleAll(master) {
191     document.querySelectorAll('.absence-item input[type=checkbox]').forEach(function(cb) {
192         cb.checked = master.checked;
193         cb.closest('.absence-item').classList.toggle('checked', master.checked);
194     });
195     updateCount();
196 }
197
198 function updateCount() {
199     var checked = document.querySelectorAll('.absence-item input[type=checkbox]:checked');
200     var n = checked.length;
201     document.getElementById('selected-count').textContent = n + (n === 1 ? ' seleccionado' : ' seleccionados');
202
203     var noteTutor = document.getElementById('note-tutor');
204     var noteInsp  = document.getElementById('note-insp');
205
206     if (n === 0) {
207         noteTutor.style.display = 'none';
208         noteInsp.style.display  = 'none';
209     } else if (n &lt;= 3) {
210         document.getElementById('lbl-t').textContent = n;
211         noteTutor.style.display = 'block';
212         noteInsp.style.display  = 'none';
213     } else {
214         document.getElementById('lbl-i').textContent = n;
215         noteTutor.style.display = 'none';
216         noteInsp.style.display  = 'block';
217     }
218
219     // Estado del master checkbox
220     var all = document.querySelectorAll('.absence-item input[type=checkbox]');
221     var masterCb = document.getElementById('select-all');
222     masterCb.indeterminate = (n &gt; 0 &amp;&amp; n &lt; all.length);
223     masterCb.checked      = (n &gt; 0 &amp;&amp; n === all.length);
224
225     checkReady();
226 }
227
228 function selectReason(el, val) {
229     document.querySelectorAll('.reason-option').forEach(function(r){ r.classList.remove('selected'); });
230     el.classList.add('selected');
231     el.querySelector('input[type=radio]').checked = true;
232
233     var otroBlock  = document.getElementById('otro-block');
234     var detailBlock = document.getElementById('detail-block');
235     var reasonText  = document.getElementById('reason-text');

```



```
236     var reasonDetail = document.getElementById('reason-detail');
237
238     if (val === 'Otro') {
239         otroBlock.style.display = 'block';
240         detailBlock.style.display = 'none';
241         reasonText.required = true;
242         reasonText.name      = 'reason';
243         reasonDetail.name    = '';
244     } else {
245         otroBlock.style.display = 'none';
246         detailBlock.style.display = 'block';
247         reasonText.required = false;
248         reasonText.name      = '';
249         reasonDetail.name    = 'reason';
250     }
251     checkReady();
252 }
253
254 function checkReady() {
255     var hasAbsence = document.querySelectorAll('.absence-item input:checked').length > 0;
256     var hasReason  = document.querySelector('input[name=reason_type]:checked');
257     document.getElementById('submitBtn').disabled = !(hasAbsence && hasReason);
258 }
259 </script>
260
261 </body>
262 </html>
```

82. ArchImp/views/notifications/index.php

```
1  <?php
2  // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3  // views/notifications/index.php
4
5  $typeIcons = [
6      'info'          => '■',
7      'success'       => '■',
8      'warning'       => '■',
9      'danger'        => '■',
10     'ausente'       => '■',
11     'justificacion' => '■',
12 ];
13 ?>
14 <!DOCTYPE html>
15 <html lang="es">
16 <head>
17     <meta charset="UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0">
19     <title>Notificaciones - EcuAsist</title>
20     <style>
21         * { box-sizing: border-box; margin: 0; padding: 0; }
22         body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif; background: #f4f6f9; color: #333; }
23
24         .breadcrumb { padding: 10px 24px; background: #fff; border-bottom: 1px solid #e0e0e0; font-size: 0.83rem; color: #000; }
25         .breadcrumb a { color: #007bff; text-decoration: none; }
26
27         .container { max-width: 760px; margin: 28px auto; padding: 0 16px; }
28
29         .page-title { font-size: 1.4rem; font-weight: 600; margin-bottom: 20px; }
30         .page-title span { color: #6c757d; font-size: 0.95rem; font-weight: 400; margin-left: 8px; }
31
32         .alert { padding: 10px 16px; border-radius: 6px; margin-bottom: 16px; font-size: 0.88rem; }
33         .alert-success { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
34
35         .btn { padding: 7px 14px; border-radius: 6px; border: none; cursor: pointer; font-size: 0.85rem; font-weight: 500; }
36         .btn:hover { opacity: 0.85; }
37         .btn-primary { background: #007bff; color: #fff; }
38         .btn-outline { background: #fff; color: #555; border: 1px solid #ccc; }
39         .btn-sm { padding: 4px 10px; font-size: 0.78rem; }
40
41         .stats-row { display: flex; gap: 12px; margin-bottom: 16px; flex-wrap: wrap; }
42         .stat-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 12px 18px; flex: 1; min-width: 280px; }
43         .stat-card .number { font-size: 1.6rem; font-weight: 700; }
44         .stat-card .label { font-size: 0.78rem; color: #888; margin-top: 2px; }
45         .stat-card.blue .number { color: #007bff; }
46
47         .notif-list { display: flex; flex-direction: column; gap: 8px; }
48
49         .notif-item {
50             background: #fff;
51             border: 1px solid #e0e0e0;
52             border-radius: 8px;
53             padding: 14px 16px;
54             display: flex;
55             gap: 12px;
56             align-items: flex-start;
57             position: relative;
58             text-decoration: none;
59         }
```

```

59         color: inherit;
60         transition: box-shadow 0.15s, border-color 0.15s;
61     }
62     .notif-item:hover { box-shadow: 0 2px 8px rgba(0,0,0,0.08); }
63     .notif-item.has-link { cursor: pointer; }
64     .notif-item.has-link:hover { border-color: #007bff; box-shadow: 0 3px 12px rgba(0,123,255,0.15); }
65     .notif-item.unread { border-left: 4px solid #007bff; background: #f8fbff; }
66
67     .notif-dot {
68         width: 8px; height: 8px; border-radius: 50%;
69         background: #007bff;
70         position: absolute; top: 14px; right: 14px;
71     }
72
73     .notif-icon { font-size: 1.5rem; min-width: 36px; text-align: center; line-height: 1; }
74     .notif-body { flex: 1; min-width: 0; }
75     .notif-title { font-weight: 600; font-size: 0.92rem; margin-bottom: 3px; }
76     .notif-message { font-size: 0.85rem; color: #555; line-height: 1.4; }
77     .notif-meta { display: flex; align-items: center; gap: 10px; margin-top: 6px; flex-wrap: wrap; }
78     .notif-time { font-size: 0.78rem; color: #999; }
79     .notif-link-hint { font-size: 0.78rem; color: #007bff; font-weight: 500; }
80     .badge-new { font-size: 0.72rem; background: #007bff; color: #fff; padding: 1px 7px; border-radius: 10px; font-w.
81
82     .notif-actions { display: flex; gap: 6px; align-items: center; flex-shrink: 0; }
83
84     .empty-state { text-align: center; padding: 60px 20px; color: #999; }
85     .empty-state .icon { font-size: 3rem; margin-bottom: 12px; }
86
87     .pagination { display: flex; justify-content: center; gap: 6px; margin-top: 24px; flex-wrap: wrap; }
88     .pagination a, .pagination span { padding: 6px 12px; border-radius: 6px; font-size: 0.85rem; border: 1px solid #.
89     .pagination a:hover { background: #f0f0f0; }
90     .pagination .current { background: #007bff; color: #fff; border-color: #007bff; }
91     </style>
92 </head>
93 <body>
94
95 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
96
97 <div class="breadcrumb">
98     <a href="?action=dashboard">■ Inicio</a> &rsquo; Notificaciones
99 </div>
100
101 <div class="container">
102
103     <?php if (isset($_GET['success'])): ?>
104         <?php $msgs = ['read' => '■ Marcadas como leídas.', 'deleted' => '■■ Notificación eliminada.', 'cleaned
105         <?php if (!empty($msgs[$_GET['success']])): ?>
106             <div class="alert alert-success"><?=$msgs[$_GET['success']] ?></div>
107         <?php endif; ?>
108     <?php endif; ?>
109
110     <div class="page-title">
111         ■ Notificaciones
112         <span><?=$total ?> en total</span>
113     </div>
114
115     <div class="stats-row">
116         <div class="stat-card blue">
117             <div class="number"><?=$unreadCount ?></div>

```

```

118         <div class="label">No leídas</div>
119     </div>
120     <div class="stat-card">
121         <div class="number"><?= $total ?></div>
122         <div class="label">Total</div>
123     </div>
124 </div>
125
126 <div style="display:flex; gap:8px; margin-bottom:16px; flex-wrap:wrap;">
127     <?php if ($unreadCount > 0): ?>
128         <form method="POST" action="?action=notifications_mark_all" style="margin:0;">
129             <button type="submit" class="btn btn-primary">■ Marcar todas como leídas</button>
130         </form>
131     <?php endif; ?>
132     <form method="POST" action="?action=notifications_delete_read"
133         onsubmit="return confirm('¿Eliminar todas las notificaciones leídas?');" style="margin:0;">
134         <button type="submit" class="btn btn-outline">■ Limpiar leídas</button>
135     </form>
136 </div>
137
138 <div class="notif-list">
139     <?php if (empty($notifications)): ?>
140         <div class="empty-state">
141             <div class="icon">■</div>
142             <p>No tienes notificaciones.</p>
143         </div>
144     <?php else: ?>
145         <?php foreach ($notifications as $n): ?>
146             <?php
147                 $icon    = $typeIcons[$n['type']] ?? '■';
148                 $isNew    = !(bool)$n['is_read'];
149                 $hasLink  = !empty($n['link']);
150                 $tag      = $hasLink ? 'a' : 'div';
151                 $href      = $hasLink ? ' href="' . htmlspecialchars($n['link']) . '" : ''';
152                 $css       = 'notif-item' . ($isNew ? ' unread' : '') . ($hasLink ? ' has-link' : '');
153                 $onclick   = ($hasLink & $isNew) ? ' onclick="markRead(' . $n['id'] . ', this)" : ''';
154             <?php
155             <?><?= $tag ?><?= $href ?> class="<?= $css ?>"<?= $onclick ?><?>
156
157             <?php if ($isNew): ?><span class="notif-dot"></span><?php endif; ?>
158
159             <div class="notif-icon"><?= $icon ?></div>
160
161             <div class="notif-body">
162                 <div class="notif-title"><?= htmlspecialchars($n['title']) ?></div>
163                 <div class="notif-message"><?= htmlspecialchars($n['message']) ?></div>
164                 <div class="notif-meta">
165                     <span class="notif-time">■ <?= date('d/m/Y H:i', strtotime($n['created_at'])) ?>
166                     <?php if ($isNew): ?><span class="badge-new">Nueva</span><?php endif; ?>
167                     <?php if ($hasLink): ?><span class="notif-link-hint">■ Clic para ver →</span>
168                 </div>
169             </div>
170
171             <div class="notif-actions" onclick="event.stopPropagation();">
172                 <?php if ($isNew): ?>
173                     <a href="?action=notifications_mark_read&id=<?= $n['id'] ?>"
174                         class="btn btn-outline btn-sm" title="Marcar como leída">✓</a>
175                 <?php endif; ?>
176                 <form method="POST" action="?action=notifications_delete" style="margin:0;"

```

```

177         onsubmit="return confirm('¿Eliminar esta notificación?');">
178         <input type="hidden" name="notification_id" value="&lt;?= $n['id'] ?>";>
179         <button type="submit" class="btn btn-outline btn-sm" title="Eliminar">■</button>
180     </form>
181 </div>
182
183     <!--<?= $tag ?>-->
184     <?php endforeach; ?>
185 <?php endif; ?>
186 </div>
187
188 <?php if ($totalPages > 1): ?>
189 <div class="pagination">
190     <?php if ($page > 1): ?>
191         <a href="action=notifications&page=&lt;?= $page - 1 ?>";>← Anterior</a>
192     <?php endif; ?>
193     <?php for ($i = 1; $i &lt;= $totalPages; $i++): ?>
194         <?php if ($i == $page): ?>
195             <span class="current">&lt;?= $i ?></span>
196         <?php else: ?>
197             <a href="action=notifications&page=&lt;?= $i ?>";>&lt;?= $i ?></a>
198         <?php endif; ?>
199     <?php endfor; ?>
200     <?php if ($page &lt; $totalPages): ?>
201         <a href="action=notifications&page=&lt;?= $page + 1 ?>";>Siguiente →</a>
202     <?php endif; ?>
203 </div>
204 <?php endif; ?>
205
206 </div>
207
208 <script>
209 function markRead(id, el) {
210     // Marcar como leída en background
211     fetch('?action=notifications_mark_read&id=' + id, { credentials: 'same-origin' })
212         .catch(function(){});
213
214     // Quitar estilos no-leída visualmente
215     el.classList.remove('unread');
216     var dot = el.querySelector('.notif-dot');
217     if (dot) dot.remove();
218     var badgeNew = el.querySelector('.badge-new');
219     if (badgeNew) badgeNew.remove();
220
221     // Actualizar campana del navbar
222     setTimeout(function(){
223         fetch('?action=notifications_unread_json', { credentials: 'same-origin' })
224             .then(function(r){ return r.json(); })
225             .then(function(data){
226                 var b = document.getElementById('notif-badge');
227                 if (b) {
228                     if (data.count > 0) {
229                         b.textContent = data.count > 99 ? '99+' : data.count;
230                         b.style.display = 'flex';
231                     } else {
232                         b.style.display = 'none';
233                     }
234                 }
235             }).catch(function(){});

```

```
236     }, 200);
237 }
238 </script>
239
240 </body>
241 </html>
```

83. ArchImp/views/partials/navbar.php

```

1 <?php
2 // views/partials/navbar.php
3
4 $current_page = $_GET['action'] ?? 'dashboard';
5
6 $_notifCount = 0;
7 if (isset($_SESSION['user_id'])) {
8     require_once BASE_PATH . '/models/Notification.php';
9     $_notifDb    = new Database();
10    $_notifModel = new Notification($_notifDb);
11    $_notifCount = $_notifModel->getUnreadCount($_SESSION['user_id']);
12 }
13
14 // Determinar qué sección está activa para auto-expandirla
15 $_activeSection = '';
16 if (in_array($current_page, ['attendance_register','attendance_view','attendance_calendar','my_attendance','tutor_course.'
17 elseif (in_array($current_page, ['my_justifications','pending_justifications','reviewed_justifications'])) $_activeSecti.
18 elseif (in_array($current_page, ['institution','academic','users','assignments','tutor_management','manage_representativ.
19 elseif (in_array($current_page, ['reports','stats'])) $_activeSection = 'reportes';
20 ?&gt;;
21 &lt;/style&gt;;
22 @import url('https://fonts.googleapis.com/css2?family=Plus+Jakarta+Sans:wght@400;500;600;700&display=swap');
23
24 :root {
25     --nav-bg: #1e40af;
26     --nav-accent: #60a5fa;
27     --nav-hover: rgba(96,165,250,0.18);
28     --nav-text: #e2e8f0;
29     --nav-muted: #bfdbfe;
30     --nav-border: rgba(255,255,255,0.12);
31     --nav-dd-bg: #ld4ed8;
32     --nav-dd-hover:#2563eb;
33     --nav-height: 52px;
34 }
35 *, ::before, ::after { box-sizing: border-box; }
36
37 /* ■■ Barra ████████████████████████████████████████ */
38 .ec-nav {
39     background: var(--nav-bg);
40     height: var(--nav-height);
41     display: flex; align-items: center;
42     position: sticky; top: 0; z-index: 1000;
43     box-shadow: 0 1px 0 var(--nav-border), 0 4px 20px rgba(0,0,0,0.25);
44     font-family: 'Plus Jakarta Sans', system-ui, sans-serif;
45 }
46 .ec-nav__inner {
47     max-width: 1400px; margin: 0 auto;
48     padding: 0 20px; width: 100%;
49     display: flex; align-items: center; gap: 4px;
50 }
51
52 /* ■■ Brand ████████████████████████████████████████ */
53 .ec-brand { display: flex; align-items: center; gap: 8px; text-decoration: none; margin-right: 12px; flex-shrink: 0; }
54 .ec-brand__logo {
55     width: 28px; height: 28px; background: #fff; border-radius: 7px;
56     display: flex; align-items: center; justify-content: center;
57     font-size: 14px; font-weight: 800; color: var(--nav-bg); letter-spacing: -0.5px;
58 }

```

[illegible]

[illegible]

```

177 DRAWER MÓVIL - acordeón
178 ████████████████████████████████████████████████████ */
179
180 /* Strip de usuario */
181 .ec-mob-user {
182     display: flex; align-items: center; gap: 12px;
183     padding: 12px 16px;
184     border-bottom: 1px solid var(--nav-border);
185     background: rgba(0,0,0,0.15);
186 }
187 .ec-mob-user__avatar {
188     width: 36px; height: 36px; border-radius: 50%;
189     background: rgba(255,255,255,0.2);
190     display: flex; align-items: center; justify-content: center;
191     font-size: 13px; font-weight: 700; color: #fff;
192     border: 2px solid rgba(255,255,255,0.3); flex-shrink: 0;
193 }
194 .ec-mob-user__name { font-size: 13px; font-weight: 600; color: #fff; flex: 1; min-width: 0; white-space: nowrap; overflow: hidden; }
195 .ec-mob-user__actions { display: flex; align-items: center; gap: 4px; flex-shrink: 0; }
196
197 /* Ítem directo (sin sub-menú) */
198 .ec-mob-direct {
199     display: flex; align-items: center; gap: 12px;
200     padding: 13px 20px;
201     color: var(--nav-text); text-decoration: none;
202     font-size: 14px; font-weight: 500;
203     border-left: 3px solid transparent;
204     transition: background 0.12s, border-color 0.12s;
205 }
206 .ec-mob-direct:hover { background: var(--nav-hover); color: #fff; }
207 .ec-mob-direct.active { background: rgba(255,255,255,0.12); color: #fff; border-left-color: #fff; font-weight: 600; }
208 .ec-mob-direct__icon { font-size: 18px; flex-shrink: 0; }
209
210 /* ■■ Acordeón ████████████████████████████████████████████████████ */
211 .ec-acc {
212     border-bottom: 1px solid rgba(255,255,255,0.07);
213 }
214
215 /* Cabecera del acordeón */
216 .ec-acc__head {
217     display: flex; align-items: center; gap: 12px;
218     padding: 13px 20px;
219     color: var(--nav-muted);
220     font-size: 13px; font-weight: 600;
221     letter-spacing: 0.01em;
222     cursor: pointer;
223     user-select: none;
224     transition: background 0.12s, color 0.12s;
225     list-style: none;
226 }
227 .ec-acc__head:hover { background: var(--nav-hover); color: #fff; }
228 .ec-acc__head.has-active { color: #fff; }
229 .ec-acc__head__icon { font-size: 18px; flex-shrink: 0; }
230 .ec-acc__head__label { flex: 1; }
231 .ec-acc__head__arrow {
232     font-size: 11px; color: var(--nav-muted);
233     transition: transform 0.22s ease;
234     margin-left: auto; flex-shrink: 0;
235     line-height: 1;

```

[illegible]

```

295         <div class="ec-item">
296             <a href="?action=dashboard" class="ec-item_btn <?php if($current_page=='dashboard') { $active=''; } else { $active='active'; } ?>"
297                 <span class="ec-item__icon"><img alt="Dashboard icon" /></span> Inicio
298             </a>
299         </div>
300
301     <?php if(Security::hasRole(['docente', 'autoridad', 'inspector', 'estudiante'])): ?>
302     <div class="ec-item">
303         <span class="ec-item_btn <?php if(in_array($current_page, ['attendance_register', 'attendance_view', 'attendance_calendar', 'my_justifications', 'pending_justifications', 'reviewed_justifications', 'institution', 'academic', 'users', 'assignments', 'tutor_management', 'manage_representatives'])) { $active=''; } else { $active='active'; } ?>"
304             <span class="ec-item__icon"><img alt="Attendance icon" /></span> Asistencia <span class="ec-item__caret"><img alt="Dropdown arrow" /></span>
305         </span>
306         <div class="ec-dropdown">
307             <?php if(Security::hasRole(['docente', 'autoridad'])): ?>
308                 <a href="?action=attendance_register" class="<?php if($current_page=='attendance_register') { $active=''; } else { $active='active'; } ?>"
309                     <?php endif; ?>
310             <?php if(Security::hasRole('docente')): ?>
311                 <a href="?action=tutor_course_attendance" class="<?php if($current_page=='tutor_course_attendance') { $active=''; } else { $active='active'; } ?>"
312                     <?php endif; ?>
313             <?php if(Security::hasRole(['docente', 'inspector', 'autoridad'])): ?>
314                 <a href="?action=attendance_view" class="<?php if($current_page=='attendance_view') { $active=''; } else { $active='active'; } ?>"
315                 <a href="?action=attendance_calendar" class="<?php if($current_page=='attendance_calendar') { $active=''; } else { $active='active'; } ?>"
316                     <?php endif; ?>
317             <?php if(Security::hasRole('estudiante')): ?>
318                 <a href="?action=my_attendance" class="<?php if($current_page=='my_attendance') { $active=''; } else { $active='active'; } ?>"
319                     <?php endif; ?>
320             </div>
321         </div>
322     <?php endif; ?>
323
324     <?php if(Security::hasRole(['estudiante', 'autoridad', 'inspector'])): ?>
325     <div class="ec-item">
326         <span class="ec-item_btn <?php if(in_array($current_page, ['my_justifications', 'pending_justifications', 'reviewed_justifications'])) { $active=''; } else { $active='active'; } ?>"
327             <span class="ec-item__icon"><img alt="Justifications icon" /></span> Justificaciones <span class="ec-item__caret"><img alt="Dropdown arrow" /></span>
328         </span>
329         <div class="ec-dropdown">
330             <?php if(Security::hasRole('estudiante')): ?>
331                 <a href="?action=my_justifications" class="<?php if($current_page=='my_justifications') { $active=''; } else { $active='active'; } ?>"
332                     <?php endif; ?>
333             <?php if(Security::hasRole(['autoridad', 'inspector'])): ?>
334                 <a href="?action=pending_justifications" class="<?php if($current_page=='pending_justifications') { $active=''; } else { $active='active'; } ?>"
335                 <a href="?action=reviewed_justifications" class="<?php if($current_page=='reviewed_justifications') { $active=''; } else { $active='active'; } ?>"
336                     <?php endif; ?>
337             </div>
338         </div>
339     <?php endif; ?>
340
341     <?php if(Security::hasRole('autoridad')): ?>
342     <div class="ec-item">
343         <span class="ec-item_btn <?php if(in_array($current_page, ['institution', 'academic', 'users', 'assignments', 'tutor_management', 'manage_representatives'])) { $active=''; } else { $active='active'; } ?>"
344             <span class="ec-item__icon"><img alt="Administration icon" /></span> Administración <span class="ec-item__caret"><img alt="Dropdown arrow" /></span>
345         </span>
346         <div class="ec-dropdown">
347             <a href="?action=institution" class="<?php if($current_page=='institution') { $active=''; } else { $active='active'; } ?>"
348             <a href="?action=academic" class="<?php if($current_page=='academic') { $active=''; } else { $active='active'; } ?>"
349             <a href="?action=users" class="<?php if($current_page=='users') { $active=''; } else { $active='active'; } ?>"
350             </div>
351             <div class="ec-dropdown_divider"></div>
352             <a href="?action=assignments" class="<?php if($current_page=='assignments') { $active=''; } else { $active='active'; } ?>"
353             <a href="?action=tutor_management" class="<?php if($current_page=='tutor_management') { $active=''; } else { $active='active'; } ?>"
354             <a href="?action=manage_representatives" class="<?php if($current_page=='manage_representatives') { $active=''; } else { $active='active'; } ?>"

```

```

354         <div class="ec-dropdown_divider"></div>
355         <a href="?action=schedules" class="<?=$current_page==='schedules'? 'active': '' ?>">■ Hor.
356         <a href="?action=backups" class="<?=$current_page==='backups'? 'active': '' ?>">■ Respa.
357     </div>
358 </div>
359
360 <div class="ec-item">
361     <span class="ec-item_btn <?=$current_page==='reports'? 'active': '' ?>">
362         <span class="ec-item_icon">■</span> Reportes <span class="ec-item_caret">▼</span>
363     </span>
364     <div class="ec-dropdown">
365         <a href="?action=reports" class="<?=$current_page==='reports'? 'active': '' ?>">■ Generar.
366         <a href="?action=stats" class="<?=$current_page==='stats'? 'active': '' ?>">■ Estadísti.
367     </div>
368 </div>
369 <?php endif; ?>
370
371 <?php if(Security::hasRole('representante')): ?>
372 <div class="ec-item">
373     <a href="?action=my_children" class="ec-item_btn <?=$current_page==='my_children'? 'active': '' ?>">
374         <span class="ec-item_icon">■■■■■■■■</span> Mis Representados
375     </a>
376 </div>
377 <?php endif; ?>
378 </div>
379
380 <!-- Right desktop -->
381 <div class="ec-right">
382     <a href="?action=notifications" class="ec-icon-btn" title="Notificaciones">
383         ■
384     <?php if ($_notifCount > 0): ?>
385         <span class="ec-notif-badge" id="notif-badge"><?=$_notifCount > 99 ? '99+' : $_notifCou.
386     <?php else: ?>
387         <span class="ec-notif-badge" id="notif-badge" style="display:none;">0</span>
388     <?php endif; ?>
389 </a>
390 <div class="ec-divider-v"></div>
391 <div class="ec-user-wrap">
392     <a href="?action=profile" class="ec-user-btn">
393         <div class="ec-user-avatar">
394             <?=$current_page==='profile'? 'active': '' ?>
395             <div>
396                 <a href="?action=profile" class="ec-user-btn">
397                     <div class="ec-user-avatar">
398                         <?=$current_page==='profile'? 'active': '' ?>
399                         <div>
400                             <?=$current_page==='profile'? 'active': '' ?>
401                             <div class="ec-user-avatar">
402                                 <?=$current_page==='profile'? 'active': '' ?>
403                                 <div>
404                                     <?=$current_page==='profile'? 'active': '' ?>
405                                     <div class="ec-user-avatar">
406                                         <?=$current_page==='profile'? 'active': '' ?>
407                                         <div>
408                                             <?=$current_page==='profile'? 'active': '' ?>
409                                             <div>
410                                                 <?=$current_page==='profile'? 'active': '' ?>
411                                                 <div class="ec-user-avatar">
412                                                     <?=$current_page==='profile'? 'active': '' ?>

```

[illegible]

```

472 <lt;/div>&gt;
473 <lt;?php endif; ?&gt;
474
475 <lt;!-- ■■ Justificaciones ■■ --&gt;
476 <lt;?php if(Security::hasRole(['estudiante','autoridad','inspector'])): ?&gt;
477 <lt;div class="ec-acc" data-section="justificaciones"&gt;
478     <lt;div class="ec-acc__head <lt;? $activeSection==='justificaciones'? 'has-active':'' ?&gt;"&gt;
479         <lt;span class="ec-acc__head__icon"&gt;■&lt;/span>&gt;
480         <lt;span class="ec-acc__head__label"&gt;Justificaciones&lt;/span>&gt;
481         <lt;span class="ec-acc__head__arrow"&gt;▼&lt;/span>&gt;
482     <lt;/div>&gt;
483     <lt;div class="ec-acc__body" style="display:none;"&gt;
484         <lt;?php if(Security::hasRole('estudiante')): ?&gt;
485             <lt;a href="?action=my_justifications" class="ec-acc__link <lt;? $current_page==='my_justifications'?'.
486             <lt;?php endif; ?&gt;
487             <lt;?php if(Security::hasRole(['autoridad','inspector'])): ?&gt;
488                 <lt;a href="?action=pending_justifications" class="ec-acc__link <lt;? $current_page==='pending_justificati.
489                 <lt;a href="?action=reviewed_justifications" class="ec-acc__link <lt;? $current_page==='reviewed_justificat.
490             <lt;?php endif; ?&gt;
491         <lt;/div>&gt;
492     <lt;/div>&gt;
493 <lt;?php endif; ?&gt;
494
495 <lt;!-- ■■ Administración ■■ --&gt;
496 <lt;?php if(Security::hasRole('autoridad')): ?&gt;
497 <lt;div class="ec-acc" data-section="administracion"&gt;
498     <lt;div class="ec-acc__head <lt;? $activeSection==='administracion'? 'has-active':'' ?&gt;"&gt;
499         <lt;span class="ec-acc__head__icon"&gt;■■&lt;/span>&gt;
500         <lt;span class="ec-acc__head__label"&gt;Administración&lt;/span>&gt;
501         <lt;span class="ec-acc__head__arrow"&gt;▼&lt;/span>&gt;
502     <lt;/div>&gt;
503     <lt;div class="ec-acc__body" style="display:none;"&gt;
504         <lt;a href="?action=institution" class="ec-acc__link <lt;? $current_page==='institution'? 'active.'.
505         <lt;a href="?action=academic" class="ec-acc__link <lt;? $current_page==='academic'? 'active':''.
506         <lt;a href="?action=users" class="ec-acc__link <lt;? $current_page==='users'? 'active':'' ?'.
507         <lt;a href="?action=assignments" class="ec-acc__link <lt;? $current_page==='assignments'? 'active.'.
508         <lt;a href="?action=tutor_management" class="ec-acc__link <lt;? $current_page==='tutor_management'? 'a.
509         <lt;a href="?action=manage_representatives" class="ec-acc__link <lt;? $current_page==='manage_representativ.
510         <lt;a href="?action=schedules" class="ec-acc__link <lt;? $current_page==='schedules'? 'active':''.
511         <lt;a href="?action=backups" class="ec-acc__link <lt;? $current_page==='backups'? 'active':''.
512     <lt;/div>&gt;
513 <lt;/div>&gt;
514
515 <lt;!-- ■■ Reportes ■■ --&gt;
516 <lt;div class="ec-acc" data-section="reportes"&gt;
517     <lt;div class="ec-acc__head <lt;? $activeSection==='reportes'? 'has-active':'' ?&gt;"&gt;
518         <lt;span class="ec-acc__head__icon"&gt;■&lt;/span>&gt;
519         <lt;span class="ec-acc__head__label"&gt;Reportes&lt;/span>&gt;
520         <lt;span class="ec-acc__head__arrow"&gt;▼&lt;/span>&gt;
521     <lt;/div>&gt;
522     <lt;div class="ec-acc__body" style="display:none;"&gt;
523         <lt;a href="?action=reports" class="ec-acc__link <lt;? $current_page==='reports'? 'active':'' ?&gt;"&gt;&lt;sp.
524         <lt;a href="?action=stats" class="ec-acc__link <lt;? $current_page==='stats'? 'active':'' ?&gt;"&gt;&lt;sp.
525     <lt;/div>&gt;
526 <lt;/div>&gt;
527 <lt;?php endif; ?&gt;
528
529 <lt;!-- Mis Representados (sin sub-menú) --&gt;
530 <lt;?php if(Security::hasRole('representante')): ?&gt;

```

```

531     <a href="?action=my_children" class="ec-mob-direct &lt;?=$current_page==='my_children'? 'active':' ' ?&gt;"&gt;
532         <span class="ec-mob-direct__icon"&gt;■■■■■■■■&lt;/span&gt; Mis Representados
533     </a&gt;
534 <?php endif; ?&gt;
535
536 <!-- Logout --&gt;
537 <a href="?action=logout" class="ec-mob-logout"&gt;
538     <svg width="16" height="16" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2.2" stroke-l
539     Cerrar Sesión
540 </a&gt;
541
542 &lt;/div&gt;&lt;!-- /#ec-mobile-drawer --&gt;
543
544 &lt;script&gt;
545 (function () {
546     /* ■■ Polling notificaciones ■■ */
547     var nb = document.getElementById('notif-badge');
548     var nbm = document.getElementById('notif-badge-mob');
549     function updateBadges(n) {
550         [nb, nbm].forEach(function(b) {
551             if (!b) return;
552             if (n &gt; 0) { b.textContent = n &gt; 99 ? '99+' : n; b.style.display = 'flex'; }
553             else b.style.display = 'none';
554         });
555     }
556     setInterval(function() {
557         fetch('?action=notifications_unread_json', { credentials: 'same-origin' })
558             .then(function(r){ return r.json(); })
559             .then(function(d){ if (typeof d.count !== 'undefined') updateBadges(d.count); })
560             .catch(function(){});
561     }, 30000);
562
563     /* ■■ Drawer ■■ */
564     var btn = document.getElementById('ec-hamburger');
565     var drawer = document.getElementById('ec-mobile-drawer');
566     if (!btn || !drawer) return;
567
568     function openDrawer() {
569         drawer.style.display = 'block';
570         btn.classList.add('open');
571         btn.setAttribute('aria-expanded', 'true');
572         document.body.style.overflow = 'hidden';
573     }
574     function closeDrawer() {
575         drawer.style.display = 'none';
576         btn.classList.remove('open');
577         btn.setAttribute('aria-expanded', 'false');
578         document.body.style.overflow = '';
579     }
580
581     btn.addEventListener('click', function() {
582         drawer.style.display === 'none' || !drawer.style.display ? openDrawer() : closeDrawer();
583     });
584
585     /* Cerrar al navegar */
586     drawer.querySelectorAll('a').forEach(function(a) {
587         a.addEventListener('click', closeDrawer);
588     });
589

```



```

590     /* Escape */
591     document.addEventListener('keydown', function(e) {
592         if (e.key === 'Escape') closeDrawer();
593     });
594
595     /* Cerrar si se pasa a desktop */
596     window.addEventListener('resize', function() {
597         if (window.innerWidth > 992) closeDrawer();
598     });
599
600     /* ■■ Acordeón ■■ */
601     var accs = drawer.querySelectorAll('.ec-acc');
602     accs.forEach(function(acc) {
603         var head = acc.querySelector('.ec-acc__head');
604         var body = acc.querySelector('.ec-acc__body');
605         if (!head || !body) return;
606
607         /* Auto-expandir si contiene el ítem activo */
608         if (head.classList.contains('has-active')) {
609             body.style.display = 'block';
610             acc.classList.add('is-open');
611         }
612
613         head.addEventListener('click', function() {
614             var isOpen = acc.classList.contains('is-open');
615
616             /* Cerrar todos los demás (comportamiento acordeón) */
617             accs.forEach(function(other) {
618                 if (other !== acc) {
619                     other.classList.remove('is-open');
620                     var ob = other.querySelector('.ec-acc__body');
621                     if (ob) ob.style.display = 'none';
622                 }
623             });
624
625             /* Toggle el actual */
626             if (isOpen) {
627                 acc.classList.remove('is-open');
628                 body.style.display = 'none';
629             } else {
630                 acc.classList.add('is-open');
631                 body.style.display = 'block';
632             }
633         });
634     });
635 }());
636 </script>

```

84. ArchImp/views/profile/change_password.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Cambiar Contraseña - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      <a href="?action=profile">Mi Perfil</a> &rsquo;
16      Cambiar Contraseña
17  </div>
18
19  <div class="container" style="max-width:500px;">
20
21      <!-- Header -->
22      <div class="page-header dark">
23          <div class="ph-icon">■</div>
24          <div>
25              <h1>Cambiar Contraseña</h1>
26              <p>Actualiza tu contraseña de acceso</p>
27          </div>
28      </div>
29
30      <?php if(isset($error)): ?>
31          <div class="alert alert-danger"><?= htmlspecialchars($error) ?></div>
32      <?php endif; ?>
33
34      <div class="panel">
35          <form method="POST">
36              <div class="form-group">
37                  <label>Contraseña Actual *</label>
38                  <input type="password" name="current_password" class="form-control" required
39                      placeholder="Ingresa tu contraseña actual">
40              </div>
41
42              <div class="form-group">
43                  <label>Nueva Contraseña *</label>
44                  <input type="password" name="new_password" class="form-control" minlength="6" required
45                      placeholder="Mínimo 6 caracteres">
46              </div>
47
48              <div class="form-group">
49                  <label>Confirmar Nueva Contraseña *</label>
50                  <input type="password" name="confirm_password" class="form-control" minlength="6" required
51                      placeholder="Repite la nueva contraseña">
52              </div>
53
54              <div style="display:flex;gap:10px;margin-top:24px;">
55                  <button type="submit" class="btn btn-primary">■ Cambiar Contraseña</button>
56                  <a href="?action=profile" class="btn btn-outline">Cancelar</a>
57              </div>
58          </form>
```

```
59      &lt;/div&gt;
60
61 &lt;/div&gt;
62 &lt;/body&gt;
63 &lt;/html&gt;
64
```

85. ArchImp/views/profile/edit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Editar Perfil - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">Inicio</a> &rsquo;
15      <a href="?action=profile">Mi Perfil</a> &rsquo;
16      Editar
17  </div>
18
19  <div class="container" style="max-width:600px;">
20
21      <!-- Header -->
22      <div class="page-header blue">
23          <div class="ph-icon"><img alt="User icon" /></div>
24          <div>
25              <h1>Editar Perfil</h1>
26              <p>Actualiza tu información personal</p>
27          </div>
28      </div>
29
30      <?php if(isset($error)): ?>
31          <div class="alert alert-danger"><?php echo htmlspecialchars($error) ?></div>
32      <?php endif; ?>
33
34      <div class="panel">
35          <form method="POST">
36              <div class="form-row">
37                  <div class="form-group">
38                      <label>Nombres *</label>
39                      <input type="text" name="first_name" class="form-control"
40                          value="<?php echo htmlspecialchars($user['first_name']) ?>" required>
41                  </div>
42                  <div class="form-group">
43                      <label>Apellidos *</label>
44                      <input type="text" name="last_name" class="form-control"
45                          value="<?php echo htmlspecialchars($user['last_name']) ?>" required>
46                  </div>
47              </div>
48
49              <div class="form-group">
50                  <label>Email *</label>
51                  <input type="email" name="email" class="form-control"
52                      value="<?php echo htmlspecialchars($user['email']) ?>" required>
53              </div>
54
55              <div class="form-group">
56                  <label>Teléfono</label>
57                  <input type="text" name="phone" class="form-control"
58                      value="<?php echo htmlspecialchars($user['phone']) ?>" placeholder="Ej: 0987654321">
59              </div>
60          </form>
61      </div>
62  </div>
63  </body>
64  </html>
```

```
59         </div>
60
61         <div style="display:flex;gap:10px;margin-top:24px;">
62             <button type="submit" class="btn btn-success">■ Guardar Cambios</button>
63             <a href="?action=profile" class="btn btn-outline">Cancelar</a>
64         </div>
65     </form>
66 </div>
67
68 </div>
69 </body>
70 </html>
71
```

86. ArchImp/views/profile/view.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mi Perfil - EcuAsist</title>
8      <style>
9          .avatar { width: 80px; height: 80px; border-radius: 50%; background: linear-gradient(135deg, #0277bd, #0288d1); }
10         .profile-header { display: flex; align-items: center; gap: 20px; margin-bottom: 24px; }
11         .info-row { display: flex; padding: 14px 0; border-bottom: 1px solid #f0f0f0; font-size: 0.9rem; }
12         .info-row:last-child { border-bottom: none; }
13         .info-label { font-weight: 600; color: #666; width: 150px; flex-shrink: 0; }
14         .info-value { color: #333; flex: 1; }
15     </style>
16 </head>
17 <body>
18
19 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
20
21 <div class="breadcrumb">
22     <a href="?action=dashboard">Inicio</a> &rsaquo;
23     Mi Perfil
24 </div>
25
26 <div class="container" style="max-width:800px;">
27
28     <?php if(isset($_GET['success'])): ?>
29         <div class="alert alert-success">✓ Perfil actualizado correctamente</div>
30     <?php endif; ?>
31     <?php if(isset($_GET['password_changed'])): ?>
32         <div class="alert alert-success">✓ Contraseña cambiada correctamente</div>
33     <?php endif; ?>
34
35     <!-- Header -->
36     <div class="page-header" style="background:linear-gradient(135deg,#0277bd,#0288d1);">
37         <div class="ph-icon">■</div>
38         <div>
39             <h1>Mi Perfil</h1>
40             <p>Información de tu cuenta</p>
41         </div>
42         <div class="ph-actions">
43             <a href="?action=edit_profile" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px .
44             <a href="?action=change_password" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1.
45         </div>
46     </div>
47
48     <div class="panel">
49         <div class="profile-header">
50             <div class="avatar">
51                 <?php $first_name = strtoupper(substr($user['first_name'], 0, 1)) . substr($user['last_name'], 0, 1); ?>
52             </div>
53             <div>
54                 <h2 style="font-size:1.3rem;font-weight:700;"><?php echo htmlspecialchars($user['first_name']) . ' ' .
55                 <p style="color:#888;margin-top:2px;"><?php echo htmlspecialchars($user['username']) ?></p>
56             </div>
57         </div>
58
```

```

59     <div class="info-row">
60         <span class="info-label">Email</span>
61         <span class="info-value">&lt;? htmlspecialchars($user['email']) ?></span>
62     </div>
63     <div class="info-row">
64         <span class="info-label">Cédula</span>
65         <span class="info-value">&lt;? htmlspecialchars($user['dni']) ?? 'No registrada' ?></span>
66     </div>
67     <div class="info-row">
68         <span class="info-label">Teléfono</span>
69         <span class="info-value">&lt;? htmlspecialchars($user['phone']) ?? 'No registrado' ?></span>
70     </div>
71     <div class="info-row">
72         <span class="info-label">Roles</span>
73         <span class="info-value">
74             &lt;?php foreach($roles as $role): ?>
75                 <span class="badge badge-green" style="margin-right:4px;">&lt;? ucfirst(htmlspecialchars($rol
76                 &lt;?php endforeach; ?>
77             </span>
78     </div>
79     <div class="info-row">
80         <span class="info-label">Miembro desde</span>
81         <span class="info-value" style="color:#888;">&lt;? date('d/m/Y', strtotime($user['created_at'])) ?&
82     </div>
83 </div>
84
85 </div>
86 </body>
87 </html>

```

87. ArchImp/views/reports/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Reportes - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo; Reportes
15  </div>
16
17  <div class="container">
18
19      <!-- Header -->
20      <div class="page-header purple">
21          <div class="ph-icon">■</div>
22          <div>
23              <h1>Generación de Reportes</h1>
24              <p>Exporta informes de asistencia en PDF o Excel</p>
25          </div>
26      </div>
27
28      <!-- Configuración -->
29      <div class="panel">
30          <h3 class="blue">■■ Configuración del Reporte</h3>
31          <div class="alert alert-info" style="margin-bottom:16px;">
32              <strong>Tipos disponibles:</strong> Vista Previa (pantalla) · PDF (formato institucional) · Exce.
33          </div>
34          <form method="POST" id="report-form">
35              <div class="form-row">
36                  <div class="form-group">
37                      <label>Curso</label>
38                      <select name="course_id" class="form-control" required>
39                          <option value="">- Seleccionar curso -</option>
40                          <?php foreach($courses as $c): ?>
41                              <option value="<? $c['id'] ?>" <? isset($course) &amp;&amp; $course['id'] == $.
42                                  <? htmlspecialchars($c['name']) ?> - <? htmlspecialchars($c['shift_name']) ?>
43                              </option>
44                          <?php endforeach; ?>
45                      </select>
46                  </div>
47                  <div class="form-group">
48                      <label>Fecha Inicio</label>
49                      <input type="date" name="start_date" class="form-control" value="<? $startDate ?? date('Y-m-0.
50                  </div>
51                  <div class="form-group">
52                      <label>Fecha Fin</label>
53                      <input type="date" name="end_date" class="form-control" value="<? $endDate ?? date('Y-m-d') ?
54                  </div>
55              </div>
56              <div style="display:flex;gap:10px;margin-top:8px;flex-wrap:wrap;">
57                  <button type="submit" name="preview" class="btn btn-primary">■■ Vista Previa</button>
58                  <button type="button" class="btn btn-danger" onclick="generateReport('pdf')" <? empty($data) ? 'd.
```



```

59         ■ Generar PDF
60         </button>;
61         <button type="button" class="btn btn-success" onclick="generateReport('excel')" &lt;?php if(!empty($data)): ?>
62             ■ Generar Excel
63         </button>;
64     </div>;
65 </form>;
66 </div>;
67
68 <!-- Vista previa -->
69 <?php if(!empty($data)): ?>
70 <?php
71 $totalPresente = count(array_filter($data, fn($d) => $d['status'] === 'presente'));
72 $totalAusente = count(array_filter($data, fn($d) => $d['status'] === 'ausente'));
73 $totalTardanza = count(array_filter($data, fn($d) => $d['status'] === 'tardanza'));
74 $totalJustificado = count(array_filter($data, fn($d) => $d['status'] === 'justificado'));
75 $total = count($data);
76 $porcentaje = $total &gt; 0 ? round(($totalPresente / $total) * 100, 1) : 0;
77 ?>
78
79 <div class="panel">
80     <h3 class="blue">■ Vista Previa del Reporte</h3>
81     <div class="alert alert-info" style="margin-bottom:14px;">
82         <strong>Curso:</strong> <?php echo htmlspecialchars($course['name']) ?> - <?php echo htmlspecialchars($course['periodo']) ?>
83         <strong>Período:</strong> <?php echo date('d/m/Y', strtotime($startDate)) ?> al <?php echo date('d/m/Y', strtotime($endDate)) ?>
84         <strong>Total:</strong> <?php echo $total ?> registros
85     </div>
86
87     <!-- Mini stats -->
88     <div class="stats-row" style="margin-bottom:16px;">
89         <div class="stat-card green">
90             <div class="number"><?php echo $totalPresente ?></div>
91             <div class="label">Presentes (<?php echo $porcentaje ?>%)</div>
92             <div class="progress-bar"><div class="progress-fill" style="width:<?php echo $porcentaje ?>%;></div></div>
93         </div>
94         <div class="stat-card red">
95             <div class="number"><?php echo $totalAusente ?></div>
96             <div class="label">Ausentes</div>
97         </div>
98         <div class="stat-card yellow">
99             <div class="number"><?php echo $totalTardanza ?></div>
100            <div class="label">Tardanzas</div>
101        </div>
102        <div class="stat-card teal">
103            <div class="number"><?php echo $totalJustificado ?></div>
104            <div class="label">Justificados</div>
105        </div>
106    </div>
107
108    <div class="table-wrap">
109        <table>
110            <thead>
111                <tr>
112                    <th>Fecha</th>
113                    <th>Estudiante</th>
114                    <th>Asignatura</th>
115                    <th>Hora</th>
116                    <th>Estado</th>
117                    <th>Observación</th>

```

```

118         </tr>
119     </thead>
120     <tbody>
121         <?php
122             $statusBadge = [
123                 'presente'    => 'badge-green',
124                 'ausente'     => 'badge-red',
125                 'tardanza'    => 'badge-yellow',
126                 'justificado' => 'badge-teal',
127             ];
128             foreach($data as $row):
129                 $bc = $statusBadge[$row['status']] ?? 'badge-gray';
130                 ?>
131                 <tr>
132                     <td><?php echo date('d/m/Y', strtotime($row['date'])) ?></td>
133                     <td><?php echo htmlspecialchars($row['student_name']) ?></td>
134                     <td><?php echo htmlspecialchars($row['subject_name']) ?></td>
135                     <td><?php echo htmlspecialchars($row['hour_period']) ?></td>
136                     <td><span class="badge <?php echo $bc ?>"><?php echo ucfirst($row['status']) ?></span>
137                     <td style="color:#888;"><?php echo htmlspecialchars($row['observation']) ?> '-' ?></td>
138                 </tr>
139             <?php endforeach; ?>
140         </tbody>
141     </table>
142 </div>
143 </div>
144
145 <?php elseif($_SERVER['REQUEST_METHOD'] === 'POST'): ?>
146 <div class="panel">
147     <div class="empty-state">
148         <div class="icon"><img alt="Empty state icon" /></div>
149         <p>No hay registros de asistencia para el período seleccionado</p>
150     </div>
151 </div>
152 <?php endif; ?>
153
154 </div>
155
156 </script>
157 function generateReport(type) {
158     var form = document.getElementById('report-form');
159     if (!form.checkValidity()) { form.reportValidity(); return; }
160     var tf = document.createElement('form');
161     tf.method = 'POST';
162     tf.style.display = 'none';
163     tf.action = type === 'pdf' ? '?action=generate_pdf' : '?action=generate_excel';
164     ['course_id', 'start_date', 'end_date'].forEach(function(n) {
165         var i = document.createElement('input');
166         i.type = 'hidden'; i.name = n;
167         i.value = form.querySelector('[name="'+n+'"]').value;
168         tf.appendChild(i);
169     });
170     document.body.appendChild(tf);
171     tf.submit();
172     document.body.removeChild(tf);
173 }
174 </script>
175
176 </body>

```

177 </html>;

178

88. ArchImp/views/representatives/child_attendance.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Asistencia - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; }
11         .navbar h1 { font-size: 24px; }
12         .navbar a { color: white; text-decoration: none; margin-left: 20px; }
13         .container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
14         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bottom: 20px; }
15         table { width: 100%; border-collapse: collapse; margin-top: 20px; }
16         th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
17         th { background: #f8f9fa; font-weight: bold; }
18         .badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
19         .badge-presente { background: #28a745; }
20         .badge-ausente { background: #dc3545; }
21         .badge-tardanza { background: #ffc107; color: #333; }
22         .badge-justificado { background: #17a2b8; }
23         .stats { display: grid; grid-template-columns: repeat(auto-fit, minmax(150px, 1fr)); gap: 15px; margin-bottom: 20px; }
24         .stat-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 20px; border-radius: 8px; }
25         .stat-number { font-size: 32px; font-weight: bold; margin-bottom: 5px; }
26         .stat-label { font-size: 13px; opacity: 0.9; }
27     </style>
28 </head>
29 <body>
30     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
31
32     <div class="container">
33         <?php
34             $total = count($attendances);
35             $presente = count(array_filter($attendances, fn($a) => $a['status'] === 'presente'));
36             $ausente = count(array_filter($attendances, fn($a) => $a['status'] === 'ausente'));
37             $tardanza = count(array_filter($attendances, fn($a) => $a['status'] === 'tardanza'));
38             $asistencia = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
39             ?>
40
41         <div class="stats">
42             <div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
43                 <div class="stat-number"><?php echo $total; ?></div>
44                 <div class="stat-label">Registros</div>
45             </div>
46             <div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
47                 <div class="stat-number"><?php echo $asistencia; ?>%</div>
48                 <div class="stat-label">Asistencia</div>
49             </div>
50             <div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
51                 <div class="stat-number"><?php echo $ausente; ?></div>
52                 <div class="stat-label">Ausencias</div>
53             </div>
54             <div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
55                 <div class="stat-number"><?php echo $tardanza; ?></div>
56                 <div class="stat-label">Tardanzas</div>
57             </div>
58         </div>
```

```

59
60     <div class="card">
61         <h2>Historial Completo</h2>
62         <table>
63             <thead>
64                 <tr>
65                     <th>Fecha</th>
66                     <th>Asignatura</th>
67                     <th>Hora</th>
68                     <th>Estado</th>
69                     <th>Observación</th>
70                 </tr>
71             </thead>
72             <tbody>
73                 <?php foreach($attendances as $att): ?>
74                     <tr>
75                         <td><?= date('d/m/Y', strtotime($att['date'])) ?></td>
76                         <td><?= $att['subject_name'] ?></td>
77                         <td><?= $att['hour_period'] ?></td>
78                         <td>
79                             <span class="badge badge-<?= $att['status'] ?>">
80                                 <?= ucfirst($att['status']) ?>
81                             </span>
82                         </td>
83                         <td><?= $att['observation'] ?: '-' ?></td>
84                     </tr>
85                 <?php endforeach; ?>
86             </tbody>
87         </table>
88     </div>
89 </div>
90 </body>
91 </html>

```

89. ArchImp/views/representatives/manage.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Gestión de Representantes - EcuAsist</title>
8      <style>
9          .rep-block { background:#fff; border:1px solid #e0e0e0; border-radius:8px; padding:16px; margin-bottom:12px; }
10         .rep-block-header { display:flex; justify-content:space-between; align-items:center; margin-bottom:12px; }
11         .rep-name { font-size:0.95rem; font-weight:700; }
12         .rep-email { font-size:0.8rem; color:#888; }
13     </style>
14 </head>
15 <body>
16
17 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
18
19 <div class="breadcrumb">
20     <a href="?action=dashboard">Inicio</a> &rsaquo;
21     Gestión de Representantes
22 </div>
23
24 <div class="container">
25
26     <?php if(isset($_GET['success'])): ?><div class="alert alert-success">✓ Relación asignada correctamente&
27     <?php if(isset($_GET['removed'])): ?><div class="alert alert-success">✓ Relación eliminada correctamente.
28     <?php if(isset($_GET['error'])): ?><div class="alert alert-danger">✗ Error al procesar la solicitud&/.
29
30     <!-- Header -->
31     <div class="page-header blue">
32         <div class="ph-icon">■</div>
33         <div>
34             <h1>Gestión de Representantes</h1>
35             <p>Vincula representantes con sus estudiantes</p>
36         </div>
37     </div>
38
39     <div style="display:grid;grid-template-columns:1fr 2fr;gap:20px;">
40
41         <!-- Formulario -->
42         <div class="panel">
43             <h3 style="margin-bottom:16px;font-size:1rem;">■ Nueva Relación</h3>
44             <form method="POST">
45                 <div class="form-group">
46                     <label>Representante *</label>
47                     <select name="representative_id" class="form-control" required>
48                         <option value="">Seleccionar representante...</option>
49                         <?php foreach($representatives as $rep): ?>
50                             <option value="<? $rep['id'] ?>">
51                                 <? $rep['last_name'] . ' ' . $rep['first_name'] ?>
52                                 (<? $rep['dni'] ?? 'Sin cédula' ?>)
53                             </option>
54                         <?php endforeach; ?>
55                     </select>
56                 </div>
57
58                 <div class="form-group">
```

```

59         <label>Estudiante </label>
60         <select name="student_id" class="form-control" required>
61             <option value="">Seleccionar estudiante...</option>
62             <?php foreach($students as $s): ?>
63                 <option value="<?php echo $s['id'];>">
64                     <?php echo htmlspecialchars($s['last_name']) . ' ' . $s['first_name'];>
65                 </option>
66             <?php endforeach; ?>
67         </select>
68     </div>
69
70     <div class="form-group">
71         <label>Parentesco </label>
72         <select name="relationship" class="form-control" required>
73             <option value="">Seleccionar...</option>
74             <option>Padre</option>
75             <option>Madre</option>
76             <option>Tutor Legal</option>
77             <option>Abuelo/a</option>
78             <option>Tío/a</option>
79             <option>Hermano/a</option>
80             <option>Otro</option>
81         </select>
82     </div>
83
84     <div class="form-group">
85         <label style="display:flex;align-items:center;gap:8px;cursor:pointer;">
86             <input type="checkbox" name="is_primary" value="1"> Representante Principal
87         </label>
88     </div>
89
90     <button type="submit" class="btn btn-success">■ Asignar Relación</button>
91 </form>
92 </div>
93
94 <!-- Listado -->
95 <div>
96     <!-- Filtros -->
97     <div class="filters" style="margin-bottom:12px;">
98         <h3 style="font-size:0.85rem;font-weight:600;color:#555;margin-bottom:8px;">■ Buscar</h3>
99         <div class="filter-grid">
100             <div>
101                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Representante.
102                 <input type="text" id="filterRep" class="form-control" placeholder="Nombre..." oninput="apply.
103             </div>
104             <div>
105                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estudiante&
106                 <input type="text" id="filterStu" class="form-control" placeholder="Nombre..." oninput="apply.
107             </div>
108             <div>
109                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Curso</lab
110                 <input type="text" id="filterCourse" class="form-control" placeholder="Curso..." oninput="app
111             </div>
112         </div>
113         <button class="btn btn-outline btn-sm" style="margin-top:8px;" onclick="clearFilters()">■■ Limpiar
114     </div>
115
116 <!-- Relaciones -->
117 <div id="rep-list">

```

```

118         <?php
119         $hasRelations = false;
120         foreach($representatives as $rep):
121             $children = $this->representativeModel->getStudentsByRepresentative($rep['id']);
122             if(!count($children)) continue;
123             $hasRelations = true;
124         ?>
125         <div class="rep-block"
126             data-repname="<?= strtolower($rep['last_name'] . ' ' . $rep['first_name']) ?>">
127             <div class="rep-block-header">
128                 <div>
129                     <div class="rep-name"><?= htmlspecialchars($rep['last_name'] . ' ' . $rep['first_
130                     <div class="rep-email"><?= htmlspecialchars($rep['email']) ?></div>
131                 </div>
132                 <span class="badge badge-blue"><?= count($children) ?> representado(s)</span>
133             </div>
134             <div class="table-wrap" style="margin-bottom:0;">
135                 <table>
136                     <thead>
137                         <tr>
138                             <th>Estudiante</th>
139                             <th>Parentesco</th>
140                             <th>Curso</th>
141                             <th>Tipo</th>
142                             <th></th>
143                         </tr>
144                     </thead>
145                     <tbody>
146                         <?php foreach($children as $child): ?>
147                         <tr class="student-row"
148                             data-student="<?= strtolower($child['last_name'] . ' ' . $child['first_name']) ?>
149                             data-course="<?= strtolower($child['course_name'] ?? '') ?>">
150                             <td><strong><?= htmlspecialchars($child['last_name'] . ' ' . $child['
151                             <td><?= htmlspecialchars($child['relationship']) ?></td>
152                             <td>
153                                 <?php if($child['course_name']): ?>
154                                     <span class="badge badge-teal"><?= htmlspecialchars($child['course_
155                                 <?php else: ?>
156                                     <span style="color:#ccc;">Sin curso</span>
157                                 <?php endif; ?>
158                             </td>
159                             <td>
160                                 <?php if($child['is_primary']): ?>
161                                     <span class="badge badge-yellow">Principal</span>
162                                 <?php else: ?>
163                                     <span class="badge badge-gray">Secundario</span>
164                                 <?php endif; ?>
165                             </td>
166                             <td>
167                                 <button class="btn btn-danger btn-sm"
168                                     onclick="confirmRemove(<?= $rep['id'] ?>, <?= $child['id'] ?>,
169                                     '<?= addslashes(htmlspecialchars($rep['last_name'] . ' ' . $rep['firs
170                                     '<?= addslashes(htmlspecialchars($child['last_name'] . ' ' . $child['
171                                 X Eliminar
172                                 </button>
173                             </td>
174                         </tr>
175                     <?php endforeach; ?>
176                 </tbody>

```



```

177         </table>
178     </div>
179 </div>
180 <?php endforeach; ?>
181
182     <?php if(!$hasRelations): ?>
183     <div class="empty-state">
184         <div class="icon">■</div>
185         <p>No hay relaciones representante-estudiante registradas.</p>
186     </div>
187     <?php endif; ?>
188 </div>
189 </div>
190 </div>
191
192 </div>
193
194 <!-- Modal confirmar eliminar -->
195 <div class="modal-overlay" id="modalRemove">
196     <div class="modal-box" style="max-width:460px;">
197         <h3>■ Eliminar Relación</h3>
198         <div id="modalRemoveBody" style="margin:12px 0;color:#555;font-size:0.88rem;">
199             <div class="modal-actions">
200                 <button class="btn btn-outline" onclick="closeModal('modalRemove')">Cancelar</button>
201                 <a id="confirmRemoveLink" href="#" class="btn btn-danger">X Sí, Eliminar</a>
202             </div>
203         </div>
204     </div>
205
206 </script>
207 function applyFilters() {
208     const rep = document.getElementById('filterRep').value.toLowerCase();
209     const stu = document.getElementById('filterStu').value.toLowerCase();
210     const course = document.getElementById('filterCourse').value.toLowerCase();
211     document.querySelectorAll('.rep-block').forEach(block => {
212         const repName = block.dataset.repname;
213         const rows = block.querySelectorAll('.student-row');
214         let vis = false;
215         rows.forEach(row => {
216             const ok = (!rep || repName.includes(rep)) &&
217                 (!stu || row.dataset.student.includes(stu)) &&
218                 (!course || row.dataset.course.includes(course));
219             row.style.display = ok ? '' : 'none';
220             if(ok) vis = true;
221         });
222         block.style.display = vis ? '' : 'none';
223     });
224 }
225 function clearFilters() {
226     ['filterRep','filterStu','filterCourse'].forEach(id => document.getElementById(id).value = '');
227     applyFilters();
228 }
229 function confirmRemove(repId, stuId, repName, stuName) {
230     document.getElementById('modalRemoveBody').innerHTML =
231         '<p>Eliminar la relación entre:</p>' +
232         '<p style="margin:10px 0;padding:10px;background:#f8f9fa;border-radius:6px;">' +
233         '<strong>Representante:</strong> ' + repName + '<br>' +
234         '<strong>Estudiante:</strong> ' + stuName + '</p>' +
235         '<p style="color:#dc3545;font-size:0.82rem;">El representante ya no podrá ver la información del estudiant.

```

```
236     document.getElementById('confirmRemoveLink').href =
237         '?action=remove_representative&rep_id=' + repId + '&student_id=' + stuId;
238     document.getElementById('modalRemove').classList.add('on');
239 }
240 function closeModal(id) { document.getElementById(id).classList.remove('on'); }
241 document.querySelectorAll('.modal-overlay').forEach(m => {
242     m.addEventListener('click', e => { if(e.target === m) closeModal(m.id); });
243 });
244 </script>
245 </body>
246 </html>
```

90. ArchImp/views/representatives/my_children.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mis Representados - EcuAsist</title>
8      <style>
9          .children-grid { display: grid; grid-template-columns: repeat(auto-fill, minmax(300px, 1fr)); gap: 16px; }
10         .child-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 10px; padding: 20px; border-top: 4px s
11         .child-card h3 { font-size: 1rem; font-weight: 700; color: #007bff; margin-bottom: 16px; }
12         .child-info { display: flex; flex-direction: column; gap: 8px; }
13         .child-row { display: flex; font-size: 0.85rem; padding: 6px 0; border-bottom: 1px solid #f0f0f0; }
14         .child-row:last-child { border-bottom: none; }
15         .child-label { font-weight: 600; color: #888; width: 100px; flex-shrink: 0; }
16         .child-val { color: #333; }
17     </style>
18 </head>
19 <body>
20
21 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
22
23 <div class="breadcrumb">
24     <a href="?action=dashboard">■ Inicio</a> &rsquo;
25     Mis Representados
26 </div>
27
28 <div class="container">
29
30     <!-- Header -->
31     <div class="page-header teal">
32         <div class="ph-icon">■■■■■■■</div>
33     </div>
34     <h1>Mis Representados</h1>
35     <p>Estudiantes vinculados a tu cuenta como representante</p>
36 </div>
37 </div>
38
39 <?php if(empty($children)): ?>
40     <div class="empty-state">
41         <div class="icon">■</div>
42         <p>No tiene estudiantes asignados como representante.</p>
43         <small style="color:#bbb;">Contacte con la autoridad para vincular a sus representados.</small>
44     </div>
45 <?php else: ?>
46     <div class="children-grid">
47         <?php foreach($children as $child): ?>
48             <div class="child-card">
49                 <h3>■ <?= htmlspecialchars($child['last_name'] . ' ' . $child['first_name']) ?>
50                 <?php if($child['is_primary']): ?>
51                     <span class="badge badge-yellow" style="font-size:0.7rem;margin-left:6px;">Principal</span>
52                 <?php endif; ?>
53                 </h3>
54                 <div class="child-info">
55                     <div class="child-row">
56                         <span class="child-label">Cédula</span>
57                         <span class="child-val">&lt;?= htmlspecialchars($child['dni']) ?? 'No registrada' ?>&lt;/span>
58                     </div>
```

```

59         <div class="child-row">
60             <span class="child-label">Parentesco</span>
61             <span class="child-val">&lt;?= htmlspecialchars($child['relationship']) ?></span>
62         </div>
63         <div class="child-row">
64             <span class="child-label">Curso</span>
65             <span class="child-val">&lt;?= htmlspecialchars($child['course_name']) ?? 'Sin asignar' ?>&
66         </div>
67         <div class="child-row">
68             <span class="child-label">Jornada</span>
69             <span class="child-val">&lt;?= $child['shift_name'] ? ucfirst($child['shift_name']) : '-' ?&
70         </div>
71     </div>
72     <div style="margin-top:16px;">
73         <a href="?action=child_attendance&student_id=&lt;?= $child['id'] ?>" class="btn btn-primary"&
74             ■ Ver Asistencia
75         </a>
76     </div>
77 </div>
78 &lt;?php endforeach; ?>
79 </div>
80 &lt;?php endif; ?>
81
82 </div>
83 </body>
84 </html>
85

```

91. ArchImp/views/schedule/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Gestión de Horarios - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      Gestión de Horarios
16  </div>
17
18  <div class="container">
19
20      <!-- Header -->
21      <div class="page-header purple">
22          <div class="ph-icon">■■</div>
23          <div>
24              <h1>Horarios de Clases</h1>
25              <p>Configure el horario semanal por curso</p>
26          </div>
27      </div>
28
29      <!-- Tabla de cursos -->
30      <div class="table-wrap">
31          <div class="table-info">
32              <span>■ <strong><?= count($courses) ?></strong> cursos disponibles</span>
33          </div>
34          <?php if(empty($courses)): ?>
35              <div class="empty-state">
36                  <div class="icon">■</div>
37                  <p>No hay cursos registrados.</p>
38              </div>
39          <?php else: ?>
40              <table>
41                  <thead>
42                      <tr>
43                          <th>Curso</th>
44                          <th>Nivel</th>
45                          <th>Paralelo</th>
46                          <th>Jornada</th>
47                          <th>Acción</th>
48                      </tr>
49                  </thead>
50                  <tbody>
51                      <?php foreach($courses as $course): ?>
52                          <tr>
53                              <td><strong><?= htmlspecialchars($course['name']) ?></strong></td>
54                              <td><?= htmlspecialchars($course['grade_level']) ?></td>
55                              <td><?= htmlspecialchars($course['parallel']) ?></td>
56                              <td><?= ucfirst($course['shift_name']) ?></td>
57                              <td>
58                                  <a href="?action=manage_schedule&course_id=<?= $course['id'] ?>"
```

```
59         class="btn btn-primary btn-sm"&gt;
60         ■■ Configurar Horario
61     </a>
62 </td>
63 </tr>
64 <?php endforeach; ?>
65 </tbody>
66 </table>
67 <?php endif; ?>
68 </div>
69
70 </div>
71 </body>
72 </html>
73
```

92. ArchImp/views/schedule/manage.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Horario de <?= htmlspecialchars($course['name']) ?> - EcuAsist</title>
8      <style>
9          .schedule-grid { display: grid; grid-template-columns: 1fr 1.5fr; gap: 20px; }
10         @media(max-width:900px) { .schedule-grid { grid-template-columns: 1fr; } }
11     </style>
12 </head>
13 <body>
14
15 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
16
17 <div class="breadcrumb">
18     <a href="?action=dashboard">■ Inicio</a> &rsquo;
19     <a href="?action=schedules">Horarios</a> &rsquo;
20     <?= htmlspecialchars($course['name']) ?>
21 </div>
22
23 <div class="container-wide">
24
25     <?php if(isset($_GET['success'])): ?>
26         <div class="alert alert-success">✓ Clase agregada al horario</div>
27     <?php endif; ?>
28     <?php if(isset($_GET['deleted'])): ?>
29         <div class="alert alert-success">✓ Clase eliminada del horario</div>
30     <?php endif; ?>
31     <?php if(isset($_GET['error'])): ?>
32         <div class="alert alert-danger">✗ <?= htmlspecialchars($_GET['error']) ?></div>
33     <?php endif; ?>
34
35     <!-- Header -->
36     <div class="page-header purple">
37         <div class="ph-icon">■</div>
38     </div>
39     <h1>Horario - <?= htmlspecialchars($course['name']) ?></h1>
40     <p>Nivel: <?= htmlspecialchars($course['grade_level']) ?> &nbsp;·&nbsp;
41     Paralelo: <?= htmlspecialchars($course['parallel']) ?> &nbsp;·&nbsp;
42     Jornada: <?= ucfirst($course['shift_name']) ?>
43     </p>
44 </div>
45 <div class="ph-actions">
46     <a href="?action=schedules" class="btn btn-outline" style="color:#fff;border-color:rgba(255,255,255,0.5);.
47 </div>
48 </div>
49
50 <div class="schedule-grid">
51
52     <!-- Formulario agregar clase -->
53     <div class="panel">
54         <h3 style="margin-bottom:16px;font-size:1rem;">■ Agregar Clase al Horario</h3>
55         <form method="POST" id="scheduleForm">
56             <div class="form-group">
57                 <label>Día de la Semana</label>
58                 <select name="day_of_week" id="day_of_week" class="form-control" required>
```

```

59         <option value="">Seleccionar...</option>;
60         <option value="lunes">Lunes</option>;
61         <option value="martes">Martes</option>;
62         <option value="miercoles">Miércoles</option>;
63         <option value="jueves">Jueves</option>;
64         <option value="viernes">Viernes</option>;
65         <option value="sabado">Sábado</option>;
66     </select>;
67 </div>;
68
69 <div class="form-group">;
70     <label>Número de Hora</label>;
71     <select name="period_number" id="period_number" class="form-control" required>;
72         <option value="">Seleccionar...</option>;
73         <?php
74             $maxHours = strpos($course['grade_level'], 'Técnico') != false ? 8 : 7;
75             for($i = 1; $i <= $maxHours; $i++): ?>;
76                 <option value="">? $i ?>"><? $i ?>ra hora</option>;
77             <?php endfor; ?>;
78     </select>;
79 </div>;
80
81 <div class="form-group">;
82     <label>Asignatura</label>;
83     <select name="subject_id" id="subject_id" class="form-control" required>;
84         <option value="">Cargando asignaturas...</option>;
85     </select>;
86 </div>;
87
88 <div class="form-group">;
89     <label>Docente Asignado</label>;
90     <input type="text" id="teacher_name" class="form-control" readonly
91         placeholder="Se asigna automáticamente" style="background:#f8f9fa;">;
92     <input type="hidden" name="teacher_id" id="teacher_id">;
93 </div>;
94
95 <div id="schedule-warning" style="display:none;" class="alert alert-warning"></div>;
96
97 <button type="submit" id="submitBtn" class="btn btn-success" disabled
98     style="opacity:0.6;cursor:not-allowed;">
99     ■ Agregar Clase
100 </button>;
101 </form>;
102 </div>;
103
104 <!-- Tabla horario actual -->;
105 <div>;
106     <div class="table-wrap">;
107         <div class="table-info">;
108             <span>■ Horario Actual – <strong><? count($schedule) ?></strong> clases</div>;
109         </div>;
110         <?php if(empty($schedule)): ?>;
111         <div class="empty-state" style="padding:30px 20px;">;
112             <div class="icon">■</div>;
113             <p>No hay clases en el horario aún.</p>;
114         </div>;
115         <?php else: ?>;
116         <table>;
117             <thead>;

```



```

118         <tr>
119             <th>Día</th>
120             <th>Hora</th>
121             <th>Asignatura</th>
122             <th>Docente</th>
123             <th></th>
124         </tr>
125     </thead>
126     <tbody>
127         <?php foreach($schedule as $class): ?>
128             <tr>
129                 <td><?= ucfirst($class['day_of_week']) ?></td>
130                 <td><?= $class['period_number'] ?>ª hora</td>
131                 <td><?= htmlspecialchars($class['subject_name']) ?></td>
132                 <td style="color:#666;"><?= htmlspecialchars($class['teacher_name']) ?></td>
133                 <td>
134                     <button class="btn btn-danger btn-sm"
135                         onclick="openModal('modalDel<?= $class['id'] ?>')">x Eliminar</button>
136                 </td>
137             </tr>
138             <?php endforeach; ?>
139         </tbody>
140     </table>
141     <?php endif; ?>
142 </div>
143 </div>
144 </div>
145
146 </div>
147
148 <!-- Modales eliminar clase -->
149 <?php foreach($schedule as $class): ?>
150 <div class="modal-overlay" id="modalDel<?= $class['id'] ?>">
151     <div class="modal-box" style="max-width:400px;">
152         <h3>■ Eliminar Clase</h3>
153         <p style="margin:12px 0 20px;color:#555;">
154             ¿Eliminar <strong><?= htmlspecialchars($class['subject_name']) ?></strong>
155             (<?= ucfirst($class['day_of_week']) ?>, <?= $class['period_number'] ?>ª hora) del horario?
156         </p>
157         <div class="modal-actions">
158             <button class="btn btn-outline" onclick="closeModal('modalDel<?= $class['id'] ?>')">Cancelar</button>
159             <form method="POST" action="?action=delete_schedule_class" style="display:inline;">
160                 <input type="hidden" name="schedule_id" value="<?= $class['id'] ?>">
161                 <input type="hidden" name="course_id" value="<?= $course['id'] ?>">
162                 <button type="submit" class="btn btn-danger">■ Eliminar</button>
163             </form>
164         </div>
165     </div>
166 </div>
167 <?php endforeach; ?>
168
169 <script>
170 const courseId = <?= (int)$course['id'] ?>;
171
172 // Cargar asignaturas
173 fetch('?action=get_course_subjects_schedule&course_id=' + courseId)
174     .then(r => r.json())
175     .then(data => {
176         const sel = document.getElementById('subject_id');

```

```

177         if (!data.length) {
178             sel.innerHTML = '<option value="">Sin asignaturas asignadas</option>';
179         } else {
180             sel.innerHTML = '<option value="">Seleccionar...</option>';
181             data.forEach(item => {
182                 const o = document.createElement('option');
183                 o.value = item.subject_id;
184                 o.textContent = item.subject_name;
185                 o.dataset.teacherId = item.teacher_id;
186                 o.dataset.teacherName = item.teacher_name;
187                 sel.appendChild(o);
188             });
189         }
190     });
191
192     document.getElementById('subject_id').addEventListener('change', function() {
193         const opt = this.options[this.selectedIndex];
194         const tid = opt.dataset.teacherId;
195         const tname = opt.dataset.teacherName;
196         document.getElementById('teacher_id').value = tid || '';
197         document.getElementById('teacher_name').value = tname || '';
198         checkReady();
199     });
200     document.getElementById('day_of_week').addEventListener('change', () => { checkConflict(); checkReady(); });
201     document.getElementById('period_number').addEventListener('change', () => { checkConflict(); checkReady(); });
202
203     function checkReady() {
204         const ok = document.getElementById('day_of_week').value &&&
205             document.getElementById('period_number').value &&&
206             document.getElementById('subject_id').value &&&
207             document.getElementById('teacher_id').value;
208         const btn = document.getElementById('submitBtn');
209         btn.disabled = !ok;
210         btn.style.opacity = ok ? '1' : '0.6';
211         btn.style.cursor = ok ? 'pointer' : 'not-allowed';
212     }
213
214     function checkConflict() {
215         const day = document.getElementById('day_of_week').value;
216         const period = document.getElementById('period_number').value;
217         if (!day || !period) return;
218         fetch('?action=check_schedule_conflict&course_id=' + courseId + '&day=' + day + '&period=' + period)
219             .then(r => r.json())
220             .then(data => {
221                 const w = document.getElementById('schedule-warning');
222                 if (data.exists) {
223                     w.textContent = '■ ■ Esta hora ya está ocupada: ' + data.subject_name + ' - ' + data.teacher_name + '. El
224                     w.style.display = 'block';
225                     document.getElementById('submitBtn').disabled = true;
226                     document.getElementById('submitBtn').style.opacity = '0.6';
227                     setTimeout(() => { w.style.display = 'none'; }, 4000);
228                 } else {
229                     w.style.display = 'none';
230                 }
231             });
232     }
233
234     function openModal(id) { document.getElementById(id).classList.add('on'); }
235     function closeModal(id) { document.getElementById(id).classList.remove('on'); }

```

```
236 document.querySelectorAll('.modal-overlay').forEach(m => {
237     m.addEventListener('click', e => { if(e.target === m) closeModal(m.id); });
238 });
239 </script>
240 </body>
241 </html>
242
```

93. ArchImp/views/stats/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Estadísticas - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">Inicio</a> &rsaquo; Estadísticas
15  </div>
16
17  <div class="container">
18
19      <!-- Header -->
20      <div class="page-header purple">
21          <div class="ph-icon"></div>
22          <div>
23              <h1>Estadísticas de Asistencia</h1>
24              <p>Análisis y métricas del período seleccionado</p>
25          </div>
26      </div>
27
28      <!-- Filtro de fechas -->
29      <div class="filters">
30          <h3>Período de análisis</h3>
31          <form method="GET">
32              <input type="hidden" name="action" value="stats">
33              <div class="filter-grid">
34                  <div class="filter-group">
35                      <label>Fecha Inicio</label>
36                      <input type="date" name="start_date" value="<?=$_GET['start_date'] ?? date('Y-m-01') ?>"&
37                  </div>
38                  <div class="filter-group">
39                      <label>Fecha Fin</label>
40                      <input type="date" name="end_date" value="<?=$_GET['end_date'] ?? date('Y-m-d') ?>"&
41                  </div>
42                  <div class="filter-group" style="display:flex;align-items:flex-end;">
43                      <button type="submit" class="btn btn-primary" style="width:100%;">Filtrar</button>
44                  </div>
45              </div>
46          </form>
47      </div>
48
49      <!-- Stats globales -->
50      <?php
51      $total = $stats['overall']['total_records'] ?? 0;
52      $pres  = $stats['overall']['presente'] ?? 0;
53      $aus   = $stats['overall']['ausente'] ?? 0;
54      $tard  = $stats['overall']['tardanza'] ?? 0;
55      $just  = $stats['overall']['justificado'] ?? 0;
56      $pct   = $total > 0 ? round(($pres / $total) * 100, 1) : 0;
57      ?>
58      <div class="stats-row">
```

```

59         <div class="stat-card gray">
60             <div class="number"><?= $total ?></div>
61             <div class="label">Total Registros</div>
62         </div>
63         <div class="stat-card green">
64             <div class="number"><?= $pres ?></div>
65             <div class="label">Presentes (<?= $pct ?>%)</div>
66             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct ?>%;background:#28.
67         </div>
68         <div class="stat-card red">
69             <div class="number"><?= $aus ?></div>
70             <div class="label">Ausentes (<?= $total > 0 ? round(($aus/$total)*100,1) : 0 ?>%)</div>
71             <div class="progress-bar"><div class="progress-fill" style="width:<?= $total>0?round(($aus/$t.
72         </div>
73         <div class="stat-card yellow">
74             <div class="number"><?= $tard ?></div>
75             <div class="label">Tardanzas</div>
76         </div>
77         <div class="stat-card teal">
78             <div class="number"><?= $just ?></div>
79             <div class="label">Justificados</div>
80         </div>
81     </div>
82
83     <!-- Asistencia por curso -->
84     <div class="panel">
85         <h3 class="blue">■ Asistencia por Curso</h3>
86         <div class="table-wrap" style="margin-top:8px;">
87             <table>
88                 <thead>
89                     <tr>
90                         <th>Curso</th>
91                         <th>Total</th>
92                         <th>Presentes</th>
93                         <th>Ausentes</th>
94                         <th>% Asistencia</th>
95                         <th style="min-width:120px;">Indicador</th>
96                     </tr>
97                 </thead>
98                 <tbody>
99                     <?php foreach($stats['by_course'] as $course):
100                         $color = $course['percentage'] >= 90 ? '#28a745' : ($course['percentage'] >= 75 ? '#ffc107.
101                     ?>
102                     <tr>
103                         <td><strong><?= htmlspecialchars($course['course_name']) ?></strong></td>
104                         <td><?= $course['total'] ?></td>
105                         <td><?= $course['presente'] ?></td>
106                         <td><?= $course['ausente'] ?></td>
107                         <td><span style="font-weight:700;color:<?= $color ?>;"><?= $course['percent.
108                         <td>
109                             <div class="progress-bar" style="height:10px;">
110                                 <div class="progress-fill" style="width:<?= $course['percentage'] ?>%;backgroun.
111                             </div>
112                         </td>
113                     </tr>
114                     <?php endforeach; ?>
115                 </tbody>
116             </table>
117         </div>

```

```

118     </div>
119
120     <!-- Top 10 ausencias -->
121     <div class="panel">
122         <h3 class="red">■ Top 10 – Estudiantes con Más Ausencias</h3>
123         <?php if(!empty($stats['by_student'])): ?>
124             <?php $maxAbs = $stats['by_student'][0]['total_absences'] ?? 1; ?>
125             <div style="margin-top:8px;">
126                 <?php foreach($stats['by_student'] as $i => $student): ?>
127                     <div style="display:flex;align-items:center;gap:10px;padding:8px 0;border-bottom:1px solid #f0f0f0;">
128                         <span style="font-size:0.75rem;color:#999;min-width:20px;text-align:right;"><?=$i+1 ?></span>
129                         <span style="flex:1;font-size:0.88rem;"><strong><?=$student['student_name'] ?></span>
130                         <span style="color:#888;font-size:0.8rem;margin-left:6px;"><?=$student['total_absences'] ?></span>
131                     </div>
132                     <div style="flex:2;">
133                         <div class="progress-bar" style="height:8px;">
134                             <div class="progress-fill" style="width:<?=$student['total_absences']/$maxAbs*100%>">
135                             </div>
136                         </div>
137                         <span style="font-weight:700;color:#dc3545;min-width:28px;text-align:right;"><?=$student['total_absences'] ?></span>
138                     </div>
139                 <?php endforeach; ?>
140             </div>
141             <?php else: ?>
142                 <p style="color:#999;font-size:0.88rem;padding:12px 0;">No hay datos de ausencias en este período.</p>
143             <?php endif; ?>
144         </div>
145
146     <!-- Tendencia diaria -->
147     <div class="panel">
148         <h3 class="blue">■ Tendencia Diaria</h3>
149         <div class="table-wrap" style="margin-top:8px;">
150             <table>
151                 <thead>
152                     <tr>
153                         <th>Fecha</th>
154                         <th>Registros</th>
155                         <th>Presentes</th>
156                         <th>% Asistencia</th>
157                         <th style="min-width:120px;">Indicador</th>
158                     </tr>
159                 </thead>
160                 <tbody>
161                     <?php foreach($stats['by_day'] as $day):
162                         $dc = $day['percentage'] >= 90 ? '#28a745' : ($day['percentage'] >= 75 ? '#ffc107' : '#dc3545');
163                     <?>
164                     <tr>
165                         <td><?=$day['date'] ?></td>
166                         <td><?=$day['total'] ?></td>
167                         <td><?=$day['presente'] ?></td>
168                         <td><?=$day['percentage'] ?></td>
169                         <td>
170                             <div class="progress-bar" style="height:8px;">
171                                 <div class="progress-fill" style="width:<?=$day['percentage'] ?>%>">
172                                 </div>
173                             </div>
174                         </td>
175                     </tr>
176                     <?php endforeach; ?>
177                 </tbody>

```

```
177         &lt;/table&gt;
178     &lt;/div&gt;
179 &lt;/div&gt;
180
181 &lt;/div&gt;
182 &lt;/body&gt;
183 &lt;/html&gt;
```

94. ArchImp/views/tutor/course_attendance.php

```
1  <?php
2  // views/tutor/course_attendance.php
3
4  $statusLabels = [
5      'presente' => ['label' => 'Presente', 'color' => '#28a745', 'bg' => '#d4edda'],
6      'ausente' => ['label' => 'Ausente', 'color' => '#dc3545', 'bg' => '#f8d7da'],
7      'tardanza' => ['label' => 'Tardanza', 'color' => '#ffc107', 'bg' => '#fff3cd'],
8      'justificado' => ['label' => 'Justificado', 'color' => '#17a2b8', 'bg' => '#d1ecf1'],
9  ];
10 $pct = function($part, $total) {
11     return $total > 0 ? round(($part / $total) * 100, 1) : 0;
12 };
13 $hasFilters = !empty($filters['subject_id']) || !empty($filters['student_id'])
14     || !empty($filters['status']) || !empty($filters['start_date'])
15     || !empty($filters['end_date']);
16 ?>
17 <!DOCTYPE html>
18 <html lang="es">
19 <head>
20     <meta charset="UTF-8">
21     <meta name="viewport" content="width=device-width, initial-scale=1.0">
22     <title>Asistencia del Curso - EcuAsist</title>
23     <style>
24         * { box-sizing: border-box; margin: 0; padding: 0; }
25         body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif; background: #f4f6f9; color: #333; }
26         .breadcrumb { padding: 10px 24px; background: #fff; border-bottom: 1px solid #e0e0e0; font-size: 0.83rem; color: #007bff; }
27         .breadcrumb a { color: #007bff; text-decoration: none; }
28         .container { max-width: 1200px; margin: 24px auto; padding: 0 16px; }
29
30         /* Page header estándar */
31         .page-header { background: linear-gradient(135deg, #007bff, #0056b3); color: #fff; border-radius: 10px; padding: 20px 24px; }
32         .page-header .ph-icon { font-size: 2.2rem; }
33         .page-header h1 { font-size: 1.3rem; font-weight: 700; }
34         .page-header p { font-size: 0.85rem; opacity: 0.85; margin-top: 4px; }
35
36         /* Stats */
37         #stats-section { display: flex; gap: 12px; margin-bottom: 20px; flex-wrap: wrap; }
38         .stat-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 14px 18px; flex: 1; min-width: 300px; }
39         .stat-card .number { font-size: 1.8rem; font-weight: 700; }
40         .stat-card .label { font-size: 0.75rem; color: #888; margin-top: 2px; text-transform: uppercase; letter-spacing: 0.5px; }
41         .stat-card .total { border-top: 1px solid #6c757d; } .stat-card .total .number { color: #6c757d; }
42         .stat-card .presente { border-top: 1px solid #28a745; } .stat-card .presente .number { color: #28a745; }
43         .stat-card .ausente { border-top: 1px solid #dc3545; } .stat-card .ausente .number { color: #dc3545; }
44         .stat-card .tardanza { border-top: 1px solid #ffc107; } .stat-card .tardanza .number { color: #e6ac00; }
45         .stat-card .justif { border-top: 1px solid #17a2b8; } .stat-card .justif .number { color: #17a2b8; }
46         .progress-bar { height: 8px; background: #e9ecef; border-radius: 4px; overflow: hidden; margin-top: 6px; }
47         .progress-fill { height: 100%; border-radius: 4px; transition: width 0.4s ease; }
48
49         /* Top ausencias */
50         .top-absences { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 16px; margin-bottom: 20px; }
51         .top-absences h3 { font-size: 0.9rem; font-weight: 600; margin-bottom: 12px; color: #dc3545; }
52         .absence-item { display: flex; align-items: center; gap: 10px; padding: 6px 0; border-bottom: 1px solid #f0f0f0; }
53         .absence-item:last-child { border-bottom: none; }
54         .absence-name { flex: 1; font-size: 0.85rem; }
55         .absence-count { font-weight: 700; color: #dc3545; font-size: 0.95rem; min-width: 28px; text-align: right; }
56         .absence-bar { flex: 2; }
57
58         /* Filtros */
```



```

59     .filters { background:#fff; border:1px solid #e0e0e0; border-radius:8px; padding:16px; margin-bottom:16px; }
60     .filters h3 { font-size:0.85rem; font-weight:600; color:#555; margin-bottom:12px; }
61     .filter-grid { display:grid; grid-template-columns:repeat(auto-fill,minmax(180px,1fr)); gap:10px; }
62     .filter-grid select,
63     .filter-grid input { width:100%; padding:7px 10px; border:1px solid #ccc; border-radius:6px; font-size:0.85rem; }
64     .filter-grid select:focus,
65     .filter-grid input:focus { border-color:#007bff; outline:none; box-shadow:0 0 2px rgba(0,123,255,0.15); }
66     .filter-grid select.on,
67     .filter-grid input.on { border-color:#007bff; background:#f0f7ff; }
68     .filter-banner { background:#fff3cd; border:1px solid #ffc107; border-radius:6px; padding:7px 12px; font-size:0.85rem; }
69     .filter-banner a { color:#856404; font-weight:700; text-decoration:none; cursor:pointer; }
70     .btn { padding:7px 16px; border-radius:6px; border:none; cursor:pointer; font-size:0.85rem; font-weight:500; text-align:center; }
71     .btn:hover { opacity:0.85; }
72     .btn-outline { background:#fff; color:#555; border:1px solid #ccc; }
73
74     /* Tabla */
75     .table-section { position:relative; }
76     .loading-overlay { display:none; position:absolute; inset:0; background:rgba(255,255,255,0.75); border-radius:8px; }
77     .loading-overlay.on { display:flex; }
78     .spinner { width:28px; height:28px; border:3px solid #e0e0e0; border-top-color:#007bff; border-radius:50%; animation:spin 1s linear infinite; }
79     @keyframes spin { to { transform:rotate(360deg); } }
80     .table-wrap { background:#fff; border:1px solid #e0e0e0; border-radius:8px; overflow:hidden; }
81     .table-info { padding:12px 16px; border-bottom:1px solid #e0e0e0; font-size:0.83rem; color:#666; display:flex; justify-content:space-between; }
82     table { width:100%; border-collapse:collapse; }
83     thead th { background:#f8f9fa; padding:10px 12px; text-align:left; font-size:0.8rem; font-weight:600; color:#555; }
84     tbody td { padding:9px 12px; border-bottom:1px solid #f0f0f0; font-size:0.85rem; }
85     tbody tr:hover { background:#f8fbff; }
86     tbody tr:last-child td { border-bottom:none; }
87     .status-badge { display:inline-block; padding:2px 10px; border-radius:12px; font-size:0.75rem; font-weight:600; }
88     .empty-state { text-align:center; padding:50px 20px; color:#999; }
89     .empty-state .icon { font-size:2.5rem; margin-bottom:10px; }
90 </style>
91 </head>
92 <body>
93
94 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
95
96 <div class="breadcrumb">
97     <a href="?action=dashboard">Inicio</a> &rsquo;
98     <a href="?action=dashboard">Mi Dashboard</a> &rsquo;
99     Asistencia de Mi Curso
100 </div>
101
102 <div class="container">
103
104     <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
105         <div class="ph-icon"><img alt="Phone icon" data-bbox="100 73 115 88"/></div>
106     </div>
107     <h1>Asistencia de Mi Curso - <?= htmlspecialchars($course['name']) ?></h1>
108     <p>
109         Jornada: <?= htmlspecialchars($course['shift_name']) ?> &nbsp; &nbsp;
110         Nivel: <?= htmlspecialchars($course['grade_level']) ?> &nbsp; &nbsp;
111         Paralelo: <?= htmlspecialchars($course['parallel']) ?>
112     </p>
113     </div>
114 </div>
115
116 <!-- Stats - actualizado por AJAX -->
117 <div id="stats-section">

```

```

118         <?php if ($stats && $stats['total'] > 0): ?>
119         <div class="stat-card total"><div class="number"><?= $stats['total'] ?></div><div cla.
120         <div class="stat-card presente">
121             <div class="number"><?= $stats['presente'] ?></div>
122             <div class="label">Presentes (<?= $pct($stats['presente'], $stats['total']) ?>%</div>
123             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct($stats['presente'], $.
124         </div>
125         <div class="stat-card ausente">
126             <div class="number"><?= $stats['ausente'] ?></div>
127             <div class="label">Ausentes (<?= $pct($stats['ausente'], $stats['total']) ?>%</div>
128             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct($stats['ausente'], $.
129         </div>
130         <div class="stat-card tardanza">
131             <div class="number"><?= $stats['tardanza'] ?></div>
132             <div class="label">Tardanzas (<?= $pct($stats['tardanza'], $stats['total']) ?>%</div>
133             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct($stats['tardanza'], $.
134         </div>
135         <div class="stat-card justif">
136             <div class="number"><?= $stats['justificado'] ?></div>
137             <div class="label">Justificados (<?= $pct($stats['justificado'], $stats['total']) ?>%</div>
138             <div class="progress-bar"><div class="progress-fill" style="width:<?= $pct($stats['justificado']).
139         </div>
140         <?php endif; ?>
141     </div>
142
143     <!-- Top ausencias (estático) -->
144     <?php if (!empty($topAbsences)): ?>
145     <div class="top-absences">
146         <h3>■ Top estudiantes con más ausencias</h3>
147         <?php $maxAbs = $topAbsences[0]['total_ausencias'] ?? 1;
148         foreach ($topAbsences as $i => $row): ?>
149             <div class="absence-item">
150                 <span style="font-size:0.75rem;color:#999;min-width:18px;"><?= $i+1 ?></span>
151                 <span class="absence-name"><?= htmlspecialchars($row['student_name']) ?></span>
152                 <div class="absence-bar"><div class="progress-bar"><div class="progress-fill" style="width:<
153                 <span class="absence-count"><?= $row['total_ausencias'] ?></span>
154             </div>
155         <?php endforeach; ?>
156     </div>
157     <?php endif; ?>
158
159     <!-- Filtros — con name para fallback, interceptados por JS -->
160     <div class="filters">
161         <h3>■ Filtrar asistencias</h3>
162         <div id="filter-banner" style="display:<?= $hasFilters ? 'flex' : 'none' ?>;" class="filter-banner">
163             <span>■ Filtros activos</span>
164             <a onclick="clearFilters()">X Limpiar filtros</a>
165         </div>
166         <div class="filter-grid">
167             <div>
168                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Asignatura</label>
169                 <select name="subject_id" id="f_subject" onchange="runFilter()" class="<?= !empty($filters['subjec.
170                     <option value="">— Todas —</option>
171                     <?php foreach ($subjects as $s): ?>
172                         <option value="<?= $s['id'] ?>" <?= ($filters['subject_id'] ?? '') == $s['id'] ? 'se.
173                         <?= htmlspecialchars($s['name']) ?>
174                     </option>
175                     <?php endforeach; ?>
176                 </select>

```

```

177         </div>
178     </div>
179     <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estudiante</label>
180     <select name="student_id" id="f_student" onchange="runFilter()" class="<? !empty($filters['student_id']) ? 'form-control' : 'form-control-sm'>">
181         <option value="">— Todos —</option>
182         <?php foreach ($students as $st):
183             <option value="<? $st['id']>" <? ($filters['student_id'] ?? '') == $st['id'] ? 'selected' : ''>
184                 <? htmlspecialchars($st['full_name'])>
185             </option>
186         <?php endforeach>
187     </select>
188 </div>
189 </div>
190 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estado</label>
191 <select name="status" id="f_status" onchange="runFilter()" class="<? !empty($filters['status']) ? 'form-control' : 'form-control-sm'>">
192     <option value="">— Todos —</option>
193     <option value="presente" <? ($filters['status'] ?? '') === 'presente' ? 'selected' : ''>
194     <option value="ausente" <? ($filters['status'] ?? '') === 'ausente' ? 'selected' : ''>
195     <option value="tardanza" <? ($filters['status'] ?? '') === 'tardanza' ? 'selected' : ''>
196     <option value="justificado" <? ($filters['status'] ?? '') === 'justificado' ? 'selected' : ''>
197 </select>
198 </div>
199 </div>
200 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Desde</label>
201 <input type="date" name="start_date" id="f_start" onchange="runFilter()"
202     value="<? htmlspecialchars($filters['start_date'] ?? '')>"
203     class="<? !empty($filters['start_date']) ? 'on' : ''>">
204 </div>
205 </div>
206 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Hasta</label>
207 <input type="date" name="end_date" id="f_end" onchange="runFilter()"
208     value="<? htmlspecialchars($filters['end_date'] ?? '')>"
209     class="<? !empty($filters['end_date']) ? 'on' : ''>">
210 </div>
211 </div>
212 </div>
213
214 <!-- Tabla — actualizada por AJAX -->
215 <div class="table-section">
216     <div class="loading-overlay" id="loader"><div class="spinner"></div></div>
217     <div class="table-wrap">
218         <div class="table-info">
219             <span><strong id="result-count"><? count($attendances)></strong> registros </span>
220             <span id="filter-tag" style="color:#856404;font-size:0.8rem;display:<? $hasFilters ? 'inline' : 'block'>:">
221         </div>
222         <div id="table-body">
223             <?php if (empty($attendances)):
224                 <div class="empty-state"><div class="icon"><img alt="Empty state icon" /></div><p>No hay registros de asistencias.</p>
225             <?php else:
226                 <table>
227                     <thead>
228                         <tr>
229                             <th>Fecha</th>
230                             <th>Hora</th>
231                             <th>Estudiante</th>
232                             <th>Asignatura</th>
233                             <th>Docente</th>
234                             <th>Estado</th>
235                         </tr>

```

```

236         &lt;td&gt;&lt;?= date('d/m/Y', strtotime($a['date'])) ?&gt;&lt;/td&gt;
237         &lt;td style="white-space:nowrap;"&gt;&lt;?= htmlspecialchars($a['hour_period']) ?&gt;&lt;/td&gt;
238         &lt;td&gt;&lt;?= htmlspecialchars($a['student_name']) ?&gt;&lt;/td&gt;
239         &lt;td&gt;&lt;?= htmlspecialchars($a['subject_name']) ?&gt;&lt;/td&gt;
240         &lt;td style="color:#666;"&gt;&lt;?= htmlspecialchars($a['teacher_name']) ?&gt;&lt;/td&gt;
241         &lt;td&gt;&lt;span class="status-badge" style="background:&lt;?= $sl['bg'] ?&gt;;color:&lt;?= $s
242         &lt;td style="color:#888;font-size:0.82rem;"&gt;&lt;?= htmlspecialchars($a['observation']) ?? '-'.
243     &lt;/tr&gt;
244     &lt;?php endforeach; ?&gt;
245     &lt;/tbody&gt;
246     &lt;/table&gt;
247     &lt;?php endif; ?&gt;
248     &lt;/div&gt;
249     &lt;/div&gt;
250 &lt;/div&gt;
251
252 &lt;/div&gt;
253
254 &lt;script&gt;
255 var SL = {
256     'presente':    { label:'Presente',    color:'#28a745', bg:'#d4edda' },
257     'ausente':    { label:'Ausente',     color:'#dc3545', bg:'#f8d7da' },
258     'tardanza':   { label:'Tardanza',     color:'#ffc107', bg:'#fff3cd' },
259     'justificado': { label:'Justificado', color:'#17a2b8', bg:'#d1ecf1' },
260 };
261
262
263 function getVals() {
264     return {
265         subject_id: document.getElementById('f_subject').value,
266         student_id: document.getElementById('f_student').value,
267         status:     document.getElementById('f_status').value,
268         start_date: document.getElementById('f_start').value,
269         end_date:   document.getElementById('f_end').value,
270     };
271 }
272
273 function hasActive(v) {
274     return v.subject_id || v.student_id || v.status || v.start_date || v.end_date;
275 }
276
277 function runFilter() {
278     var v = getVals();
279
280     // Marcar campos activos visualmente
281     ['f_subject','f_student','f_status','f_start','f_end'].forEach(function(id){
282         var el = document.getElementById(id);
283         el.classList.toggle('on', !!el.value);
284     });
285
286     // Mostrar/ocultar banner y tag
287     var active = hasActive(v);
288     document.getElementById('filter-banner').style.display = active ? 'flex' : 'none';
289     document.getElementById('filter-tag').style.display    = active ? 'inline' : 'none';
290
291     // Spinner
292     document.getElementById('loader').classList.add('on');
293
294     // Construir URL para el endpoint AJAX

```

```

295     var p = new URLSearchParams({ action: 'tutor_course_attendance_ajax' });
296     if (v.subject_id) p.set('subject_id', v.subject_id);
297     if (v.student_id) p.set('student_id', v.student_id);
298     if (v.status) p.set('status', v.status);
299     if (v.start_date) p.set('start_date', v.start_date);
300     if (v.end_date) p.set('end_date', v.end_date);
301
302     fetch('? ' + p.toString(), { credentials: 'same-origin' })
303     .then(function(r){ return r.json(); })
304     .then(function(data){
305         renderStats(data.stats, data.total);
306         renderTable(data.attendances);
307         document.getElementById('loader').classList.remove('on');
308     })
309     .catch(function(){
310         document.getElementById('loader').classList.remove('on');
311     });
312 }
313
314 function renderStats(s, total) {
315     var sec = document.getElementById('stats-section');
316     if (!s) return;
317     var t = parseInt(total) || 0;
318     var pre = parseInt(s.presente) || 0;
319     var aus = parseInt(s.ausente) || 0;
320     var tar = parseInt(s.tardanza) || 0;
321     var jus = parseInt(s.justificado) || 0;
322     function pct(v){ return t > 0 ? Math.round(v/t*1000)/10 : 0; }
323     function bar(v,c){ return '<div class="progress-bar"><div class="progress-fill" style="width:'+pct(v)+'%;ba
324     sec.innerHTML =
325         '<div class="stat-card total"><div class="number">'+t+'</div><div class="label">Total re.
326         '<div class="stat-card presente"><div class="number">'+pre+'</div><div class="label">Pre.
327         '<div class="stat-card ausente"><div class="number">'+aus+'</div><div class="label">Ause.
328         '<div class="stat-card tardanza"><div class="number">'+tar+'</div><div class="label">Tar.
329         '<div class="stat-card justif"><div class="number">'+jus+'</div><div class="label">Justi.
330 }
331
332 function renderTable(rows) {
333     document.getElementById('result-count').textContent = rows.length;
334     if (!rows.length) {
335         document.getElementById('table-body').innerHTML =
336             '<div class="empty-state"><div class="icon">■</div><p>No se encontraron registros co
337         return;
338     }
339     var html = '<table><thead><tr><th>Fecha</th><th>Hora</th><th>Estudia.
340     rows.forEach(function(a){
341         var sl = SL[a.status] || { label:a.status, color:'#666', bg:'#eee' };
342         var d = a.date ? a.date.substr(8,2)+'/'+a.date.substr(5,2)+'/'+a.date.substr(0,4) : '';
343         html += '<tr>' +
344             '<td>'+d+'</td>' +
345             '<td style="white-space:nowrap;">'+esc(a.hour_period)+'</td>' +
346             '<td>'+esc(a.student_name)+'</td>' +
347             '<td>'+esc(a.subject_name)+'</td>' +
348             '<td style="color:#666;">'+esc(a.teacher_name)+'</td>' +
349             '<td><span class="status-badge" style="background:'+sl.bg+';color:'+sl.color+';">'+sl.label+'&
350             '<td style="color:#888;font-size:0.82rem;">'+(a.observation ? esc(a.observation) : '-')+'</td>' .
351             '</tr>';
352     });
353     html += '</tbody></table>';

```

```
354     document.getElementById('table-body').innerHTML = html;
355 }
356
357 function esc(s){
358     if (!s) return '';
359     return String(s).replace(/&/g,'&amp;').replace(/</g,'&lt;').replace(/>/g,'&gt;').replace(/"/g,...
360 }
361
362 function clearFilters(){
363     ['f_subject','f_student','f_status'].forEach(function(id){ document.getElementById(id).value = ''; });
364     document.getElementById('f_start').value = '';
365     document.getElementById('f_end').value = '';
366     runFilter();
367 }
368 </script>
369
370 </body>
371 </html>
```

95. ArchImp/views/tutor/dashboard.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Dashboard Tutor - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">Inicio</a> &rsquo; Dashboard Tutor
15  </div>
16
17  <div class="container">
18
19      <?php
20      $pct = ($stats['total'] > 0)
21          ? round(($stats['presente'] + $stats['justificado']) / $stats['total'] * 100)
22          : 0;
23      $todayPct = ($todayStats['total'] > 0)
24          ? round(($todayStats['presente'] + $todayStats['justificado']) / $todayStats['total'] * 100)
25          : 0;
26      ?>
27
28      <!-- Header -->
29      <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
30          <div class="ph-icon"></div>
31          <div>
32              <h1>Dashboard - Tutor</h1>
33              <p><?php echo htmlspecialchars($course['name']) ?> &nbsp;|&nbsp; <?php echo count($students) ?> </p>
34          </div>
35          <div class="ph-actions">
36              <a href="?action=tutor_search_students" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;bo
37              <a href="?action=tutor_course_attendance" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;
38          </div>
39      </div>
40
41      <!-- Métricas globales del curso -->
42      <div style="display:grid;grid-template-columns:repeat(5,1fr);gap:16px;margin-bottom:20px;">
43          <div class="stat-card">
44              <div class="stat-icon" style="background:#e3f2fd;"></div>
45              <div class="stat-value"><?php echo $stats['total'] ?? 0 ?></div>
46              <div class="stat-label">Total registros</div>
47          </div>
48          <div class="stat-card">
49              <div class="stat-icon" style="background:#e8f5e9;"></div>
50              <div class="stat-value" style="color:#2e7d32;"><?php echo $stats['presente'] ?? 0 ?></div>
51              <div class="stat-label">Presentes</div>
52          </div>
53          <div class="stat-card">
54              <div class="stat-icon" style="background:#ffebee;"></div>
55              <div class="stat-value" style="color:#c62828;"><?php echo $stats['ausente'] ?? 0 ?></div>
56              <div class="stat-label">Ausentes</div>
57          </div>
58          <div class="stat-card">
```

```

59         <div class="stat-icon" style="background:#ffde7;">■</div>
60         <div class="stat-value" style="color:#f57f17;"><?=$stats['tardanza'] ?? 0 ></div>
61         <div class="stat-label">Tardanzas</div>
62     </div>
63     <div class="stat-card">
64         <div class="stat-icon" style="background:#e0f7fa;">■</div>
65         <div class="stat-value" style="color:#00838f;"><?=$stats['justificado'] ?? 0 ></div>
66         <div class="stat-label">Justificados</div>
67     </div>
68 </div>
69
70 <!-- Grid: asistencia general + hoy -->
71 <div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">
72
73     <!-- Asistencia general -->
74     <div class="panel">
75         <h3 style="font-size:.95rem;margin-bottom:12px;color:#555;">■ Asistencia General del Curso</h3>
76         <div style="display:flex;justify-content:space-between;margin-bottom:6px;">
77             <span style="font-size:.85rem;color:#666;">Efectividad (presentes + justificados)</span>
78             <strong style="color:<?=$pct>=90?'#2e7d32':(<?=$pct>=75?'#f57f17':'#c62828') ><?"><?=$pct>=
79         </div>
80         <div style="background:#f0f0f0;border-radius:6px;height:12px;overflow:hidden;margin-bottom:16px;">
81             <div style="height:100%;border-radius:6px;width:<?=$pct %>background:<?=$pct>=90?'#4c
82         </div>
83         <?php if($stats['total'] > 0): >
84         <div style="display:grid;grid-template-columns:1fr 1fr;gap:8px;font-size:13px;">
85             <?php
86             $bars = [
87                 ['presente', '#4caf50', '✓ Presente'],
88                 ['ausente', '#f44336', 'X Ausente'],
89                 ['tardanza', '#ff9800', '■ Tardanza'],
90                 ['justificado', '#00acc1', '■ Justificado'],
91             ];
92             foreach($bars as [$key, $color, $label]):
93                 $n = $stats[$key] ?? 0;
94                 $pct_bar = round($n / $stats['total'] * 100);
95             >
96             <div>
97                 <div style="display:flex;justify-content:space-between;margin-bottom:3px;">
98                     <span><?=$label ></span>
99                     <span style="font-weight:700;color:<?=$color ><?"><?=$n > (<?=$pct_bar %
100                 </div>
101                 <div style="background:#f0f0f0;border-radius:4px;height:6px;">
102                     <div style="height:100%;border-radius:4px;width:<?=$pct_bar %>background:<?=$col
103                 </div>
104             </div>
105             <?php endforeach; >
106         </div>
107         <?php else: >
108         <p style="color:#aaa;font-size:13px;text-align:center;margin-top:8px;">&Sin registros de asistencia aún.
109         <?php endif; >
110     </div>
111
112 <!-- Hoy -->
113 <div class="panel">
114     <h3 style="font-size:.95rem;margin-bottom:12px;color:#555;">■ Hoy — <?=$date('d/m/Y') ></h3>
115     <?php if($todayStats['total'] > 0): >
116     <div style="display:flex;justify-content:space-between;margin-bottom:6px;">
117         <span style="font-size:.85rem;color:#666;">Asistencia del día</span>

```



```

118         <strong style="color:<?=$todayPct>=90?'#2e7d32':($todayPct>=75?'#f57f17':'#c62828') ?>"&
119     </div>
120 <div style="background:#f0f0f0;border-radius:6px;height:12px;overflow:hidden;margin-bottom:16px;">
121     <div style="height:100%;border-radius:6px;width:<?=$todayPct ?>%;background:<?=$todayPct&
122 </div>
123 <div style="display:flex;gap:8px;justify-content:center;flex-wrap:wrap;">
124     <div style="text-align:center;padding:10px 14px;background:#e8f5e9;border-radius:8px;">
125         <div style="font-size:1.4rem;font-weight:700;color:#2e7d32;"><?=$todayStats['presente'] ?&
126         <div style="font-size:11px;color:#666;">Presentes</div>
127     </div>
128     <div style="text-align:center;padding:10px 14px;background:#ffebee;border-radius:8px;">
129         <div style="font-size:1.4rem;font-weight:700;color:#c62828;"><?=$todayStats['ausente'] ?&
130         <div style="font-size:11px;color:#666;">Ausentes</div>
131     </div>
132     <div style="text-align:center;padding:10px 14px;background:#fffde7;border-radius:8px;">
133         <div style="font-size:1.4rem;font-weight:700;color:#f57f17;"><?=$todayStats['tardanza'] ?&
134         <div style="font-size:11px;color:#666;">Tardanzas</div>
135     </div>
136     <div style="text-align:center;padding:10px 14px;background:#e0f7fa;border-radius:8px;">
137         <div style="font-size:1.4rem;font-weight:700;color:#00838f;"><?=$todayStats['justificado']>
138         <div style="font-size:11px;color:#666;">Justificados</div>
139     </div>
140 </div>
141 <?php else: ?>
142 <div style="text-align:center;padding:30px 0;color:#aaa;">
143     <div style="font-size:2rem;margin-bottom:8px;">■</div>
144     <p style="font-size:13px;">No hay registros de asistencia hoy</p>
145 </div>
146 <?php endif; ?>
147 </div>
148 </div>
149
150 <!-- Grid: tendencia + top ausencias -->
151 <div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">
152
153     <!-- Tendencia 7 días -->
154     <div class="panel">
155         <h3 style="font-size:.95rem;margin-bottom:16px;color:#555;">■ Últimos 7 días</h3>
156         <?php if(!empty($trend)): ?>
157         <div style="display:flex;align-items:flex-end;gap:8px;height:80px;">
158             <?php foreach($trend as $day):
159                 $dp = ($day['total'] > 0) ? round($day['presente'] / $day['total'] * 100) : 0;
160                 $h = max(4, round($dp * 0.8));
161                 $col = $dp>=90?'#4caf50':($dp>=75?'#ff9800':'#f44336');
162                 ?>
163                 <div style="flex:1;display:flex;flex-direction:column;align-items:center;gap:4px;">
164                     <span style="font-size:10px;color:#999;"><?=$dp ?>%</span>
165                     <div style="width:100%;height:<?=$h ?>px;background:<?=$col ?>;border-radius:3px 3px;
166                     <span style="font-size:10px;color:#999;"><?=$date('d/m', strtotime($day['date'])) ?></span>
167                 </div>
168             <?php endforeach; ?>
169         </div>
170         <?php else: ?>
171         <p style="color:#aaa;font-size:13px;text-align:center;padding:20px 0;">Sin datos de los últimos 7 días.
172         <?php endif; ?>
173     </div>
174
175     <!-- Top ausencias -->
176     <div class="panel">

```

```

177         <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px;">
178             <h3 style="font-size:.95rem;color:#555;">■ Estudiantes con más ausencias</h3>
179             <a href="?action=tutor_search_students" style="font-size:12px;color:#007bff;text-decoration:none;"&
180         </div>
181         <?php if(!empty($stopAbsences)): ?>
182         <?php foreach($stopAbsences as $i => $row): ?>
183         <div style="display:flex;align-items:center;gap:10px;padding:8px 0;border-bottom:1px solid #f5f5f5;">
184             <span style="font-size:11px;color:#999;width:16px;text-align:right;"><? $i+1 ?></span>
185             <div style="flex:1;font-size:13px;color:#333;"><? htmlspecialchars($row['student_name']) ?>
186             <span class="badge <? $row['total_ausencias']>=10?'badge-red':($row['total_ausencias']>=5?'
187                 <? $row['total_ausencias'] ?> faltas
188             </span>
189         </div>
190         <?php endforeach; ?>
191         <?php else: ?>
192         <p style="color:#aaa;font-size:13px;text-align:center;padding:20px 0;">■ Sin ausencias registradas<
193         <?php endif; ?>
194     </div>
195 </div>
196
197 <!-- Lista de estudiantes del curso -->
198 <div class="table-wrap">
199     <div class="table-info">
200         <span>■ <strong>Estudiantes del Curso</strong> - <? count($students) ?></span>
201         <a href="?action=tutor_search_students" class="btn btn-primary btn-sm">■ Buscar estudiante</a>
202     </div>
203     <?php if(empty($students)): ?>
204     <div class="empty-state" style="padding:30px;">
205         <div class="icon">■</div>
206         <p>No hay estudiantes matriculados en este curso.</p>
207     </div>
208     <?php else: ?>
209     <table>
210         <thead>
211             <tr><th>#</th><th>Estudiante</th></tr>
212         </thead>
213         <tbody>
214             <?php $i = 1; foreach($students as $s): ?>
215             <tr>
216                 <td style="color:#999;font-size:12px;"><? $i++ ?></td>
217                 <td><? htmlspecialchars($s['full_name']) ?></td>
218             </tr>
219             <?php endforeach; ?>
220         </tbody>
221     </table>
222     <?php endif; ?>
223 </div>
224
225 </div>
226 </body>
227 </html>

```

96. ArchImp/views/tutor/no_tutor.php

```
1  <?php // views/tutor/no_tutor.php ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Asistencia de Mi Curso - EcuAsist</title>
7      <style>
8          * { margin:0; padding:0; box-sizing:border-box; }
9          body { font-family: Arial, sans-serif; background:#f4f4f4; }
10         .container { max-width:1200px; margin:24px auto; padding:0 16px; }
11         .page-header { background:linear-gradient(135deg,#1565c0,#1976d2); color:#fff; border-radius:10px; padding:20px .
12         .page-header .ph-icon { font-size:2.2rem; }
13         .page-header h1 { font-size:1.3rem; font-weight:700; }
14         .page-header p { font-size:0.85rem; opacity:0.85; margin-top:4px; }
15         .empty-state { background:white; border-radius:10px; padding:60px 20px; text-align:center; box-shadow:0 2px 10px.
16         .empty-state .icon { font-size:3.5rem; margin-bottom:16px; }
17         .empty-state h2 { font-size:1.3rem; color:#333; margin-bottom:8px; }
18         .empty-state p { color:#888; font-size:14px; margin-bottom:24px; }
19         .btn-back { display:inline-block; padding:10px 22px; background:#007bff; color:#fff; border-radius:6px; text-dec
20         .btn-back:hover { background:#0056b3; }
21     </style>
22 </head>
23 <body>
24
25 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
26
27 <div class="breadcrumb">
28     <a href="?action=dashboard">Inicio &rsaquo; Asistencia de Mi Curso
29 </div>
30
31 <div class="container">
32
33     <div class="page-header">
34         <div class="ph-icon">■</div>
35         <div>
36             <h1>Asistencia de Mi Curso</h1>
37             <p>Vista del docente tutor sobre la asistencia de su curso</p>
38         </div>
39     </div>
40
41     <div class="empty-state">
42         <div class="icon">■■■</div>
43         <h2>No eres tutor de ningún curso</h2>
44         <p>Actualmente no tienes asignado un curso como tutor en el año lectivo activo.</p>
45         <a href="?action=dashboard" class="btn-back">← Volver al inicio</a>
46     </div>
47
48 </div>
49 </body>
50 </html>
```

97. ArchImp/views/tutor/search_students.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Buscar Estudiantes - EcuAsist</title>
8  </head>
9  <body>
10
11  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
12
13  <div class="breadcrumb">
14      <a href="?action=dashboard">■ Inicio</a> &rsquo;
15      <a href="?action=tutor_dashboard">Dashboard Tutor</a> &rsquo;
16      Buscar Estudiantes
17  </div>
18
19  <div class="container">
20
21      <!-- Header -->
22      <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
23          <div class="ph-icon">■</div>
24          <div>
25              <h1>Buscar Estudiantes</h1>
26              <p><?= htmlspecialchars($course['name']) ?></p>
27          </div>
28          <div class="ph-actions">
29              <a href="?action=tutor_dashboard" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1.
30              </div>
31          </div>
32
33      <!-- Buscador -->
34      <div class="panel" style="margin-bottom:20px;">
35          <form method="GET" action="">
36              <input type="hidden" name="action" value="tutor_search_students">
37              <div class="form-row" style="align-items:flex-end;">
38                  <div class="form-group" style="flex:1;">
39                      <label>Nombre, apellido o cédula</label>
40                      <input type="text" name="q" class="form-control"
41                          placeholder="Ej: García, 1750..."
42                          value="<?= htmlspecialchars($_GET['q']) ?? '' ?>"
43                          autofocus>
44                  </div>
45                  <div class="form-group" style="flex:0 0 auto;display:flex;gap:8px;">
46                      <button type="submit" class="btn btn-primary">■ Buscar</button>
47                      <?php if(!empty($_GET['q'])): ?>
48                          <a href="?action=tutor_search_students" class="btn btn-outline">✕ Limpiar</a>
49                      <?php endif; ?>
50                  </div>
51              </div>
52          </form>
53      </div>
54
55      <!-- Resultados -->
56      <?php if(isset($_GET['q']) && empty($students)): ?>
57          <div class="empty-state">
58              <div class="icon">■</div>
```

```

59         <p>No se encontraron estudiantes con "<?php echo htmlspecialchars($_GET['q']);>"</p>
60     </div>
61
62     <?php else: ?>
63     <div class="table-wrap">
64         <div class="table-info">
65             <span>
66                 ■ <strong><?php if(isset($_GET['q']) && $_GET['q'] != '' ? 'Resultados' : 'Todos los estudi
67                 - <?php count($students) ?> encontrado<?php count($students)!=1?'s':'' ?>
68                 <?php if(!empty($_GET['q'])): ?>
69                     para "<strong><?php echo htmlspecialchars($_GET['q']);>"</strong>"
70                 <?php endif; ?>
71             </span>
72         </div>
73         <?php if(empty($students)): ?>
74         <div class="empty-state" style="padding:30px;">
75             <div class="icon">■</div>
76             <p>No hay estudiantes matriculados en este curso.</p>
77         </div>
78         <?php else: ?>
79         <table>
80             <thead>
81                 <tr>
82                     <th>#</th>
83                     <th>Estudiante</th>
84                     <th>Cédula</th>
85                     <th>Email</th>
86                     <th>Teléfono</th>
87                     <th style="text-align:center;">Clases</th>
88                     <th style="text-align:center;">■</th>
89                     <th style="text-align:center;">■</th>
90                     <th style="text-align:center;">■</th>
91                     <th style="text-align:center;">% Asistencia</th>
92                 </tr>
93             </thead>
94             <tbody>
95                 <?php $i = 1; foreach($students as $s):
96                     $total = (int)($s['total_clases'] ?? 0);
97                     $pres = (int)($s['presentes'] ?? 0);
98                     $ause = (int)($s['ausentes'] ?? 0);
99                     $tard = (int)($s['tardanzas'] ?? 0);
100                    $just = (int)($s['justificados'] ?? 0);
101                    $pct = $total > 0 ? round(($pres + $just) / $total * 100) : 0;
102                    $pctColor = $pct >= 90 ? '#2e7d32' : ($pct >= 75 ? '#f57f17' : '#c62828');
103                    $pctBg = $pct >= 90 ? '#e8f5e9' : ($pct >= 75 ? '#fff8e1' : '#ffebee');
104                    ?>
105                 <tr>
106                     <td style="color:#999;font-size:12px;"><?php echo $i++ ?></td>
107                     <td><strong><?php echo htmlspecialchars($s['last_name']) . ' ' . $s['first_name'] ?></td>
108                     <td style="font-size:13px;color:#666;"><?php echo htmlspecialchars($s['dni']) ?? '-' ?></td>
109                     <td style="font-size:13px;color:#666;"><?php echo htmlspecialchars($s['email']) ?? '-' ?></td>
110                     <td style="font-size:13px;color:#666;"><?php echo htmlspecialchars($s['phone']) ?? '-' ?></td>
111                     <td style="text-align:center;font-size:13px;"><?php echo $total ?></td>
112                     <td style="text-align:center;">
113                         <span style="font-weight:700;color:#2e7d32;"><?php echo $pres ?></span>
114                     </td>
115                     <td style="text-align:center;">
116                         <span style="font-weight:700;color:#c62828;"><?php echo $ause ?></span>
117                     </td>

```

```
118         &lt;td style="text-align:center;"&gt;
119             &lt;span style="font-weight:700;color:#f57f17;"&gt;&lt;?= $tard ?&gt;&lt;/span&gt;
120         &lt;/td&gt;
121         &lt;td style="text-align:center;"&gt;
122             &lt;?php if($total &gt; 0): ?&gt;
123                 &lt;span style="background:&lt;?= $pctBg ?&gt;;color:&lt;?= $pctColor ?&gt;;font-weight:700;padd.
124                     &lt;?= $pct ?&gt;%
125                 &lt;/span&gt;
126             &lt;?php else: ?&gt;
127                 &lt;span style="color:#ccc;font-size:12px;"&gt;Sin datos&lt;/span&gt;
128             &lt;?php endif; ?&gt;
129         &lt;/td&gt;
130     &lt;/tr&gt;
131     &lt;?php endforeach; ?&gt;
132 &lt;/tbody&gt;
133 &lt;/table&gt;
134 &lt;?php endif; ?&gt;
135 &lt;/div&gt;
136 &lt;?php endif; ?&gt;
137
138 &lt;/div&gt;
139 &lt;/body&gt;
140 &lt;/html&gt;
```

98. ArchImp/views/users/create.php

```

1 <?php
2 require_once BASE_PATH . '/helpers/Security.php';
3 Security::requireLogin();
4 if (!Security::hasRole('autoridad')) die('Acceso denegado');
5 ?>
6 <!DOCTYPE html>
7 <html lang="es">
8 <head>
9     <meta charset="UTF-8">
10    <meta name="viewport" content="width=device-width, initial-scale=1.0">
11    <title>Crear Usuario - EcuAsist</title>
12    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
13    <style>
14        body { background: #f4f6f9; padding-top: 0; }
15        .container { max-width: 860px; }
16        .form-section { background: #fff; border: 1px solid #e8ecf0; border-radius: 10px; padding: 22px 26px; margin-bot.
17        .form-section h5 { color: #1a237e; border-bottom: 2px solid #e8ecf0; padding-bottom: 10px; margin-bottom: 18px; }
18        .required::after { content: " *"; color: #dc3545; }
19        .form-control:focus { border-color: #1a237e; box-shadow: 0 0 3px rgba(26,35,126,0.1); }
20        .form-control.is-valid { border-color: #28a745; }
21        .form-control.is-invalid { border-color: #dc3545; }
22        .feedback-icon { position: absolute; right: 10px; top: 50%; transform: translateY(-50%); font-size: 14px; pointer.
23        .input-wrap { position: relative; }
24        .field-hint { font-size: 11.5px; color: #888; margin-top: 3px; }
25        .field-error { font-size: 12px; color: #dc3545; margin-top: 3px; display: none; }
26        .field-error.show { display: block; }
27        #passport-row { display: none; }
28        .doc-toggle { display: flex; align-items: center; gap: 10px; margin-bottom: 10px; }
29        .doc-toggle label { margin: 0; font-size: 13px; color: #555; cursor: pointer; user-select: none; }
30    </style>
31 </head>
32 <body>
33
34 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
35
36 <div class="breadcrumb">
37     <a href="?action=dashboards">Inicio</a> &rsquo;
38     <a href="?action=users">Gestión de Usuarios</a> &rsquo;
39     Crear Usuario
40 </div>
41
42 <div class="container mt-4">
43
44 <?php if (!empty($errors)): ?>
45     <div class="alert alert-danger">
46         <strong>Errores</strong>
47         <ul class="mb-0 mt-1">
48             <?php foreach ($errors as $e): ?>
49                 <li><?= htmlspecialchars($e) ?></li>
50             <?php endforeach; ?>
51         </ul>
52     </div>
53 <?php endif; ?>
54
55 <form method="POST" action="?action=create_user" id="createUserForm" novalidate>
56
57     <!-- Datos de Acceso -->
58     <div class="form-section">

```

[illegible]

[illegible]

```

177         </div>
178     <?php endforeach; ?>
179 </div>
180 </div>
181
182 <!-- Botones -->
183 <div class="d-flex justify-content-between mb-5">
184     <a href="?action=users" class="btn btn-secondary"><- Cancelar</a>
185     <button type="submit" class="btn btn-primary px-4">✓ Crear Usuario</button>
186 </div>
187
188 </form>
189 </div>
190
191 <script>
192 // Validación cédula Ecuador (algoritmo oficial)
193 function validarCedula(cedula) {
194     if (!/^d{10}$/.test(cedula)) return false;
195     var prov = parseInt(cedula.substring(0, 2));
196     if (prov < 1 || prov > 24) return false;
197     var dig = cedula.split('').map(Number);
198     var coef = [2,1,2,1,2,1,2,1,2];
199     var suma = 0;
200     for (var i = 0; i < 9; i++) {
201         var r = dig[i] * coef[i];
202         suma += r % 10 ? r - 9 : r;
203     }
204     var residuo = suma % 10;
205     var digVer = residuo === 0 ? 0 : 10 - residuo;
206     return dig[9] === digVer;
207 }
208
209 // Validación teléfono Ecuador
210 // Celular: 09XXXXXXXXX (10 dígitos, empieza en 09)
211 // Fijo: 0[2-7]XXXXXXX (9 dígitos: código provincia + 7 locales)
212 function validarTelefono(tel) {
213     var limpio = tel.replace(/[\s-]/g, '');
214     if (/^09\d{8}$/.test(limpio)) return { ok: true, tipo: 'Celular' };
215     if (/^0[2-7]\d{7}$/.test(limpio)) return { ok: true, tipo: 'Fijo' };
216     return { ok: false, tipo: null };
217 }
218
219 // Validación email
220 function validarEmail(email) {
221     return /^[^\s@]+@[^\s@]+\.[^\s@]{2,}$/ .test(email.trim());
222 }
223
224 // Helpers UI
225 function setValid(inputId, iconId, errId) {
226     var el = document.getElementById(inputId);
227     el.classList.remove('is-invalid'); el.classList.add('is-valid');
228     if (iconId) document.getElementById(iconId).textContent = '■';
229     if (errId) { document.getElementById(errId).textContent = ''; document.getElementById(errId).classList.remove('show'); }
230 }
231 function setInvalid(inputId, iconId, errId, msg) {
232     var el = document.getElementById(inputId);
233     el.classList.remove('is-valid'); el.classList.add('is-invalid');
234     if (iconId) document.getElementById(iconId).textContent = '■';
235     if (errId) { document.getElementById(errId).textContent = msg; document.getElementById(errId).classList.add('show'); }

```

[illegible]

[illegible]

99. ArchImp/views/users/edit.php

[illegible]

```

59         </ul>
60     </div>
61 <?php endif; ?>
62
63 <form method="POST" action="?action=edit_user&id=<?=$user['id'] ?>">
64     <!-- Sección Datos de Acceso -->
65     <div class="form-section">
66         <h5>■ Datos de Acceso</h5>
67         <div class="row">
68             <div class="col-md-6 mb-3">
69                 <label class="form-label required">Nombre de Usuario</label>
70                 <input type="text"
71                     name="username"
72                     class="form-control"
73                     value="<?=$htmlspecialchars($user['username']) ?>"
74                     required>
75                 <small class="text-muted">Sin espacios ni caracteres especiales</small>
76             </div>
77             <div class="col-md-6 mb-3">
78                 <label class="form-label required">Email</label>
79                 <input type="email"
80                     name="email"
81                     class="form-control"
82                     value="<?=$htmlspecialchars($user['email']) ?>"
83                     required>
84             </div>
85         </div>
86
87         <div class="password-info">
88             <strong>■ Contraseña:</strong> Deja en blanco si no deseas cambiarla
89         </div>
90
91         <div class="row">
92             <div class="col-md-6 mb-3">
93                 <label class="form-label">Nueva Contraseña</label>
94                 <input type="password"
95                     name="password"
96                     class="form-control"
97                     minlength="6">
98                 <small class="text-muted">Mínimo 6 caracteres</small>
99             </div>
100             <div class="col-md-6 mb-3">
101                 <label class="form-label">Confirmar Nueva Contraseña</label>
102                 <input type="password"
103                     name="confirm_password"
104                     class="form-control"
105                     minlength="6">
106             </div>
107         </div>
108     </div>
109
110     <!-- Sección Datos Personales -->
111     <div class="form-section">
112         <h5>■■■ Datos Personales</h5>
113         <div class="row">
114             <div class="col-md-6 mb-3">
115                 <label class="form-label required">Nombres</label>
116                 <input type="text"
117                     name="first_name"

```

```

118         class="form-control"
119         value="{<?= htmlspecialchars($user['first_name']) ?>}"
120         required&gt;
121     </div&gt;
122     <div class="col-md-6 mb-3"&gt;
123         <label class="form-label required"&gt;Apellidos</label&gt;
124         <input type="text"
125             name="last_name"
126             class="form-control"
127             value="{<?= htmlspecialchars($user['last_name']) ?>}"
128             required&gt;
129     </div&gt;
130     <div class="col-md-6 mb-3"&gt;
131         <label class="form-label"&gt;Cédula/DNI</label&gt;
132         <input type="text"
133             name="dni"
134             class="form-control"
135             value="{<?= htmlspecialchars($user['dni']) ?? ''} ?>";&gt;
136     </div&gt;
137     <div class="col-md-6 mb-3"&gt;
138         <label class="form-label"&gt;Teléfono</label&gt;
139         <input type="text"
140             name="phone"
141             class="form-control"
142             value="{<?= htmlspecialchars($user['phone']) ?? ''} ?>";&gt;
143     </div&gt;
144 </div&gt;
145 </div&gt;
146
147 <!-- Información de Roles --&gt;
148 <div class="alert alert-info"&gt;
149     <strong&gt;■■ Roles:</strong&gt; Los roles se gestionan desde la tabla de usuarios
150 </div&gt;
151
152 <!-- Botones --&gt;
153 <div class="d-flex justify-content-between mt-4"&gt;
154     <a href="?action=users" class="btn btn-secondary"&gt;
155         ← Cancelar
156     </a&gt;
157     <button type="submit" class="btn btn-success"&gt;
158         ✓ Guardar Cambios
159     </button&gt;
160 </div&gt;
161 </form&gt;
162 </div&gt;
163 </div&gt;
164 </div&gt;
165 </div&gt;
166 </div&gt;
167
168 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"&gt;</script&gt;
169 </body&gt;
170 </html&gt;

```


100. ArchImp/views/users/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Gestión de Usuarios - EcuAsist</title>
7      <style>
8          * { margin: 0; padding: 0; box-sizing: border-box; }
9          body { font-family: Arial, sans-serif; background: #f4f4f4; }
10         .container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
11         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
12         table { width: 100%; border-collapse: collapse; margin-top: 20px; }
13         th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
14         th { background: #f8f9fa; font-weight: bold; }
15         .badge {
16             display: inline-block;
17             background: #28a745;
18             color: white;
19             padding: 3px 8px;
20             border-radius: 3px;
21             font-size: 11px;
22             margin: 2px;
23             position: relative;
24         }
25         .btn-remove-role {
26             background: none;
27             border: none;
28             color: white;
29             font-size: 16px;
30             font-weight: bold;
31             cursor: pointer;
32             padding: 0;
33             margin-left: 5px;
34             line-height: 1;
35         }
36         .btn-remove-role:hover {
37             color: #ff0000;
38         }
39         select, button { padding: 8px 12px; margin: 5px; border: 1px solid #ddd; border-radius: 4px; }
40         button { background: #007bff; color: white; border: none; cursor: pointer; }
41         button:hover { background: #0056b3; }
42         .btn-danger { background: #dc3545; }
43         .btn-danger:hover { background: #c82333; }
44         .btn-success { background: #28a745; }
45         .btn-success:hover { background: #218838; }
46         .btn-warning { background: #ffc107; color: #212529; }
47         .btn-warning:hover { background: #e0a800; }
48         .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
49         .error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
50         .info { background: #d1ecf1; color: #0c5460; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
51         h2 { margin-bottom: 20px; color: #333; }
52         .header-actions {
53             display: flex;
54             justify-content: space-between;
55             align-items: center;
56             margin-bottom: 20px;
57         }
58         .btn-sm {
```

```

59         padding: 5px 10px;
60         font-size: 12px;
61         margin: 2px;
62     }
63     .action-buttons {
64         white-space: nowrap;
65     }
66     </style>
67 </head>
68 <body>
69     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
70 <div class="breadcrumb">
71     <a href="?action=dashboard">Inicio</a> &rsaquo; Gestión de Usuarios
72 </div>
73
74 <div class="container">
75
76     <div class="page-header" style="background:linear-gradient(135deg,#2e7d32,#388e3c);">
77         <div class="ph-icon"></div>
78     </div>
79     <h1>Gestión de Usuarios</h1>
80     <p>Administración de docentes, estudiantes y representantes</p>
81 </div>
82 </div>
83
84 <div class="card">
85     <?php if(isset($_GET['created'])): ?>
86         <div class="success">✔ Usuario creado correctamente</div>
87     <?php endif; ?>
88     <?php if(isset($_GET['updated'])): ?>
89         <div class="success">✔ Usuario actualizado correctamente</div>
90     <?php endif; ?>
91     <?php if(isset($_GET['deleted'])): ?>
92         <div class="success">✔ Usuario eliminado correctamente</div>
93     <?php endif; ?>
94     <?php if(isset($_GET['deactivated'])): ?>
95         <div class="info">■■ Usuario desactivado (tiene registros de asistencia)</div>
96     <?php endif; ?>
97     <?php if(isset($_GET['success'])): ?>
98         <div class="success">✔ Rol asignado correctamente</div>
99     <?php endif; ?>
100    <?php if(isset($_GET['removed'])): ?>
101        <div class="success">✔ Rol eliminado correctamente</div>
102    <?php endif; ?>
103    <?php if(isset($_GET['error']) && $_GET['error'] == 'has_assignments'): ?>
104        <div class="error">✗ No se puede eliminar el rol docente porque tiene asignaciones activas. Elimin.
105    <?php endif; ?>
106    <?php if(isset($_GET['error']) && $_GET['error'] == 'not_found'): ?>
107        <div class="error">✗ Usuario no encontrado</div>
108    <?php endif; ?>
109    <?php if(isset($_GET['error']) && $_GET['error'] == 'self_delete'): ?>
110        <div class="error">✗ No puedes eliminar tu propia cuenta</div>
111    <?php endif; ?>
112    <?php if(isset($_GET['error']) && $_GET['error'] == 'delete_failed'): ?>
113        <div class="error">✗ Error al eliminar el usuario</div>
114    <?php endif; ?>
115
116    <div class="header-actions">
117        <h2>■ Gestión de Usuarios</h2>

```

```

118         <a href="?action=create_user" style="padding: 10px 20px; background: #28a745; color: white; text-deco.
119         + Crear Usuario
120     </a>
121 </div>
122
123 <!-- Filtro por Rol - Botones -->
124 <?php
125 $roleIcons = ['docente'=>'■■■■', 'estudiante'=>'■■■■', 'inspector'=>'■', 'autoridad'=>'■■', 'representante'=>'■■■■'];
126 $roleColors = [
127     'docente' => ['active'=>'background:#007bff;color:white;border-color:#007bff;', 'outline:#007bff;'],
128     'estudiante' => ['active'=>'background:#28a745;color:white;border-color:#28a745;', 'outline:#28a745;'],
129     'inspector' => ['active'=>'background:#fd7e14;color:white;border-color:#fd7e14;', 'outline:#fd7e14;'],
130     'autoridad' => ['active'=>'background:#6f42c1;color:white;border-color:#6f42c1;', 'outline:#6f42c1;'],
131     'representante'=> ['active'=>'background:#20c997;color:white;border-color:#20c997;', 'outline:#20c997;'],
132 ];
133 $currentFilter = $_GET['filter_role'] ?? '';
134 ?>
135 <div style="display:flex;gap:8px;flex-wrap:wrap;margin-bottom:20px;">
136     <a href="?action=users"
137         style="padding:7px 16px;border-radius:20px;text-decoration:none;font-size:13px;font-weight:600;border:1px solid #343a40;
138         <?= $currentFilter==='' ? 'background:#343a40;color:white;border-color:#343a40;' : 'color:#343a40;background:white;border-color:#343a40;'"
139         ■ Todos
140     </a>
141     <?php foreach($roles as $role):
142         $rn = $role['name'];
143         $ico = $roleIcons[$rn] ?? '■';
144         $col = $roleColors[$rn] ?? ['active'=>'background:#007bff;color:white;border-color:#007bff;', 'outline:#007bff;'],
145         $isActive = ($currentFilter === $rn);
146     ?>
147     <a href="?action=users&filter_role=<?= $rn ?>"
148         style="padding:7px 16px;border-radius:20px;text-decoration:none;font-size:13px;font-weight:600;border:1px solid #343a40;
149         <?= $isActive ? $col['active'] : $col['outline']. 'background:white;' ?>">
150         <?= $ico ?> <?= ucfirst($rn) ?>
151     </a>
152     <?php endforeach; ?>
153 </div>
154
155 <table>
156     <thead>
157         <tr>
158             <th>#</th>
159             <th>Nombre Completo</th>
160             <th>Email</th>
161             <th>Roles Actuales</th>
162             <th>Asignar Rol</th>
163             <th>Acciones</th>
164         </tr>
165     </thead>
166     <tbody>
167         <?php
168         // Aplicar filtro
169         $filteredUsers = $users;
170
171         if(isset($_GET['filter_role']) && $_GET['filter_role'] != '') {
172             $filterRole = $_GET['filter_role'];
173             $filteredUsers = array_filter($users, function($user) use ($filterRole) {
174                 if($user['roles']) {
175                     $userRoles = explode(',', $user['roles']);
176                     return in_array($filterRole, $userRoles);
177                 }
178             });
179         }
180     </tbody>
181 </table>

```

```

177         }
178         return false;
179     });
180 }
181 ?>
182
183 <?php if(empty($filteredUsers)): ?>
184     <tr>
185         <td colspan="7" style="text-align:center;">No hay usuarios con este rol</td>
186     </tr>
187 <?php else: ?>
188     <?php
189     $counter = 1;
190     foreach($filteredUsers as $user):
191         ?>
192     <tr>
193         <td><?php $counter++ ?></td>
194         <td><?php $user['last_name'] . ' ' . $user['first_name'] ?></td>
195         <td><?php $user['email'] ?></td>
196         <td>
197             <?php if($user['roles']): ?>
198                 <?php
199                 // Obtener IDs de roles con nombres
200                 $db = new Database();
201                 $sqlRoles = "SELECT r.id, r.name
202                             FROM roles r
203                             INNER JOIN user_roles ur ON r.id = ur.role_id
204                             WHERE ur.user_id = :user_id
205                             ORDER BY r.name";
206                 $stmtRoles = $db->connect()->prepare($sqlRoles);
207                 $stmtRoles->execute([':user_id' => $user['id']]);
208                 $userRolesData = $stmtRoles->fetchAll(PDO::FETCH_ASSOC);
209
210                 foreach($userRolesData as $roleData):
211                     ?>
212                     <span class="badge">
213                         <?php $roleData['name'] ?>
214                         <form method="POST" action="?action=remove_role" ?>
215                             <input type="hidden" name="user_id" value="<?php $user['id'] ?>">
216                             <input type="hidden" name="role_id" value="<?php $roleData['id'] ?>">
217                             <button type="submit" class="btn-remove-role">x</button>
218                         </form>
219                     </span>
220                 <?php endforeach; ?>
221             <?php else: ?>
222                 <em>Sin roles</em>
223             <?php endif; ?>
224         </td>
225     </td>
226     <form method="POST" action="?action=assign_role" ?>
227         <input type="hidden" name="user_id" value="<?php $user['id'] ?>">
228         <select name="role_id" required style="padding: 5px;">
229             <option value="">Seleccionar...</option>
230             <?php foreach($roles as $role): ?>
231                 <option value="<?php $role['id'] ?>"><?php $role['name'] ?>
232             <?php endforeach; ?>
233         </select>
234         <button type="submit" style="padding: 5px 10px;">Asignar</button>
235     </form>

```

```

236         </td>
237         <td class="action-buttons">
238             <a href="?action=edit_user&id=<?=$user['id'] ?>"
239                 class="btn-warning btn-sm"
240                 style="text-decoration: none; padding: 5px 10px; border-radius: 4px;">
241                 =■ Editar
242             </a>
243             <form method="POST" action="?action=delete_user" style="display: inline;" onsubmit="r.
244                 <input type="hidden" name="user_id" value="<?=$user['id'] ?>">
245                 <button type="submit" class="btn-danger btn-sm">
246                 ■■ Eliminar
247             </button>
248         </form>
249     </td>
250 </tr>
251 <?php endforeach; ?>
252 <?php endif; ?>
253 </tbody>
254 </table>
255 </div>
256 </div>
257
258 <script>
259 function confirmDelete(event, userName) {
260     event.preventDefault();
261
262     const modal = document.createElement('div');
263     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);
264
265     const modalContent = document.createElement('div');
266     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow:
267
268     modalContent.innerHTML = `
269         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■■ Eliminar Usuario</h3>
270         <p style="margin: 0 0 20px 0; color: #666;">
271             ¿Está seguro de eliminar al usuario <strong>${userName}</strong>?
272         </p>
273         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">
274             <strong>Nota:</strong> Si el usuario tiene asistencias registradas, será desactivado en luga.
275         </p>
276         <div style="display: flex; gap: 10px; justify-content: flex-end;">
277             <button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: whi.
278                 Cancelar
279             </button>
280             <button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: whi.
281                 Sí, Eliminar
282             </button>
283         </div>
284     `;
285
286     modal.appendChild(modalContent);
287     document.body.appendChild(modal);
288
289     const form = event.target;
290
291     document.getElementById('confirmDeleteBtn').onclick = function() {
292         document.body.removeChild(modal);
293         form.submit();
294     };

```

```

295
296     document.getElementById('cancelDeleteBtn').onclick = function() {
297         document.body.removeChild(modal);
298     };
299
300     return false;
301 }
302
303 function confirmRemoveRole(event, roleName, userName) {
304     event.preventDefault();
305
306     const modal = document.createElement('div');
307     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);';
308
309     const modalContent = document.createElement('div');
310     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow: 0 0 10px #000;';
311
312     modalContent.innerHTML = `
313         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ Eliminar Rol</h3>
314         <p style="margin: 0 0 20px 0; color: #666;">
315             ¿Está seguro de eliminar el rol <strong>${roleName}</strong> del usuario <strong>${userName}</strong>?
316         </p>
317         <div style="display: flex; gap: 10px; justify-content: flex-end;">
318             <button type="button" id="cancelRemoveRoleBtn" style="padding: 10px 20px; background: #6c757d; color: white;">
319                 Cancelar
320             </button>
321             <button type="button" id="confirmRemoveRoleBtn" style="padding: 10px 20px; background: #dc3545; color: white;">
322                 Sí, Eliminar
323             </button>
324         </div>
325     `;
326
327     modal.appendChild(modalContent);
328     document.body.appendChild(modal);
329
330     const form = event.target;
331
332     document.getElementById('confirmRemoveRoleBtn').onclick = function() {
333         document.body.removeChild(modal);
334         form.submit();
335     };
336
337     document.getElementById('cancelRemoveRoleBtn').onclick = function() {
338         document.body.removeChild(modal);
339     };
340
341     return false;
342 }
343 </script>
344 </body>
345 </html>

```