

# **DOCUMENTACIÓN DEL PROYECTO**

## **Sistema de Asistencia Escolar**

Generado: 18/02/2026 23:04

# ÍNDICE DE ARCHIVOS

N°	Archivo	Ruta
1	ARCHITECTURE.md	/
2	backup_2026-02-17_23-49-43.sql	ArchImp/backups
3	config.php	ArchImp/config
4	database.php	ArchImp/config
5	AcademicController.php	ArchImp/controllers
6	AssignmentController.php	ArchImp/controllers
7	AttendanceController.php	ArchImp/controllers
8	AuthController.php	ArchImp/controllers
9	BackupController.php	ArchImp/controllers
10	DashboardController.php	ArchImp/controllers
11	ExportController.php	ArchImp/controllers
12	InstitutionController.php	ArchImp/controllers
13	JustificationController.php	ArchImp/controllers
14	NotificationController.php	ArchImp/controllers
15	ProfileController.php	ArchImp/controllers
16	ReportController.php	ArchImp/controllers
17	ReportController2.php	ArchImp/controllers
18	RepresentativeController.php	ArchImp/controllers
19	ScheduleController.php	ArchImp/controllers
20	SearchController.php	ArchImp/controllers
21	StatsController.php	ArchImp/controllers
22	TutorController.php	ArchImp/controllers
23	UserController.php	ArchImp/controllers
24	Backup.php	ArchImp/helpers
25	Logger.php	ArchImp/helpers
26	Mailer.php	ArchImp/helpers
27	Security.php	ArchImp/helpers
28	Attendance.php	ArchImp/models
29	ClassSchedule.php	ArchImp/models
30	Course.php	ArchImp/models
31	Institution.php	ArchImp/models
32	InstitutionShift.php	ArchImp/models
33	Justification.php	ArchImp/models
34	Notification.php	ArchImp/models
35	Representative.php	ArchImp/models

36	Role.php	ArchImp/models
37	SchoolYear.php	ArchImp/models
38	Shift.php	ArchImp/models
39	Subject.php	ArchImp/models
40	TeacherAssignment.php	ArchImp/models
41	User.php	ArchImp/models
42	index.php	ArchImp/public
43	course_edit.php	ArchImp/views/academic
44	course_students.php	ArchImp/views/academic
45	create.php	ArchImp/views/academic
46	edit.php	ArchImp/views/academic
47	enroll.php	ArchImp/views/academic
48	index.php	ArchImp/views/academic
49	subject_edit.php	ArchImp/views/academic
50	index.php	ArchImp/views/assignments
51	tutor.php	ArchImp/views/assignments
52	view_course.php	ArchImp/views/assignments
53	calendar.php	ArchImp/views/attendance
54	course_view.php	ArchImp/views/attendance
55	my_attendance.php	ArchImp/views/attendance
56	register.php	ArchImp/views/attendance
57	view.php	ArchImp/views/attendance
58	forgot.php	ArchImp/views/auth
59	login.php	ArchImp/views/auth
60	register.php	ArchImp/views/auth
61	reset.php	ArchImp/views/auth
62	index.php	ArchImp/views/backup
63	index.php	ArchImp/views/dashboard
64	index.php	ArchImp/views/institution
65	my_list.php	ArchImp/views/justifications
66	pending.php	ArchImp/views/justifications
67	reviewed.php	ArchImp/views/justifications
68	submit.php	ArchImp/views/justifications
69	index.php	ArchImp/views/notifications
70	navbar.php	ArchImp/views/partials
71	change_password.php	ArchImp/views/profile
72	edit.php	ArchImp/views/profile
73	view.php	ArchImp/views/profile
74	index.php	ArchImp/views/reports

75	child_attendance.php	ArchImp/views/representatives
76	manage.php	ArchImp/views/representatives
77	my_children.php	ArchImp/views/representatives
78	index.php	ArchImp/views/schedule
79	manage.php	ArchImp/views/schedule
80	index.php	ArchImp/views/stats
81	course_attendance.php	ArchImp/views/tutor
82	no_tutor.php	ArchImp/views/tutor
83	create.php	ArchImp/views/users
84	edit.php	ArchImp/views/users
85	index.php	ArchImp/views/users
86	FuncionalidadesRol.md	/
87	README.md	/
88	ResumenEjecutivo_06C.md	/
89	SugerenciasFuturas.md	/
90	composer.json	/
91	mallá.md	/
92	men#U00fa_v2.0.md	/

# ESTRUCTURA DE CARPETAS

FilesImport/

ArchImp/

views/

assignments/

auth/

tutor/

users/

dashboard/

schedule/

profile/

backup/

stats/

notifications/

partials/

representatives/

reports/

attendance/

academic/

institution/

justifications/

helpers/

config/

public/

models/

controllers/

backups/

# CÓDIGO FUENTE

## 1. ARCHITECTURE.md

Ruta: 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. backup\_2026-02-17\_23-49-43.sql

Ruta: ArchImp/backups/backup\_2026-02-17\_23-49-43.sql

```
1  -- MariaDB dump 10.19  Distrib 10.4.32-MariaDB, for Win64 (AMD64)
2  --
3  -- Host: localhost    Database: ecuasistencia2026_db
4  --
5  -- Server version    10.4.32-MariaDB
6
7  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!40101 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `activity_logs`
20 --
21
22 DROP TABLE IF EXISTS `activity_logs`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!40101 SET character_set_client = utf8 */;
25 CREATE TABLE `activity_logs` (
26   `id` int(11) NOT NULL AUTO_INCREMENT,
27   `user_id` int(11) NOT NULL,
28   `action` varchar(100) NOT NULL,
29   `entity_type` varchar(50) DEFAULT NULL,
30   `entity_id` int(11) DEFAULT NULL,
31   `description` text DEFAULT NULL,
32   `ip_address` varchar(45) DEFAULT NULL,
33   `user_agent` varchar(255) DEFAULT NULL,
34   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
35   PRIMARY KEY (`id`),
36   KEY `idx_user` (`user_id`),
37   KEY `idx_date` (`created_at`),
38   CONSTRAINT `activity_logs_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
39 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
40 /*!40101 SET character_set_client = @saved_cs_client */;
41
42 --
43 -- Dumping data for table `activity_logs`
44 --
45
46 LOCK TABLES `activity_logs` WRITE;
47 /*!40000 ALTER TABLE `activity_logs` DISABLE KEYS */;
48 /*!40000 ALTER TABLE `activity_logs` ENABLE KEYS */;
49 UNLOCK TABLES;
50
51 --
52 -- Table structure for table `attendances`
53 --
54
55 DROP TABLE IF EXISTS `attendances`;
56 /*!40101 SET @saved_cs_client      = @@character_set_client */;
57 /*!40101 SET character_set_client = utf8 */;
58 CREATE TABLE `attendances` (
59   `id` int(11) NOT NULL AUTO_INCREMENT,
60   `student_id` int(11) NOT NULL,
61   `course_id` int(11) NOT NULL,
62   `subject_id` int(11) NOT NULL,
63   `teacher_id` int(11) NOT NULL,
64   `school_year_id` int(11) NOT NULL,
65   `shift_id` int(11) NOT NULL,
66   `date` date NOT NULL,
67   `hour_period` varchar(20) DEFAULT NULL,
68   `status` enum('presente','ausente','tardanza','justificado') NOT NULL,
69   `observation` text DEFAULT NULL,
70   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
71   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
72   PRIMARY KEY (`id`),
73   KEY `course_id` (`course_id`),
74   KEY `subject_id` (`subject_id`),
75   KEY `teacher_id` (`teacher_id`),
```

```

76 KEY `school_year_id` (`school_year_id`),
77 KEY `shift_id` (`shift_id`),
78 KEY `idx_date` (`date`),
79 KEY `idx_student` (`student_id`),
80 CONSTRAINT `attendances_ibfk_1` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),

81 CONSTRAINT `attendances_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
82 CONSTRAINT `attendances_ibfk_3` FOREIGN KEY (`subject_id`) REFERENCES `subjects` (`id`),
83 CONSTRAINT `attendances_ibfk_4` FOREIGN KEY (`teacher_id`) REFERENCES `users` (`id`),
84 CONSTRAINT `attendances_ibfk_5` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`),
85 CONSTRAINT `attendances_ibfk_6` FOREIGN KEY (`shift_id`) REFERENCES `shifts` (`id`)
86 ) ENGINE=InnoDB AUTO_INCREMENT=49 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
87 /*!40101 SET character_set_client = @saved_cs_client */;
88
89 --
90 -- Dumping data for table `attendances`
91 --
92
93 LOCK TABLES `attendances` WRITE;
94 /*!40000 ALTER TABLE `attendances` DISABLE KEYS */;
95 INSERT INTO `attendances` VALUES (1,8,1,1,2,1,1,'2026-02-07','1ra hora','presente','','2026-02-11 23:56:54','2026-02-11 ...
96 /*!40000 ALTER TABLE `attendances` ENABLE KEYS */;
97 UNLOCK TABLES;
98
99 --
100 -- Table structure for table `class_schedule`
101 --
102
103 DROP TABLE IF EXISTS `class_schedule`;
104 /*!40101 SET @saved_cs_client = @@character_set_client */;
105 /*!40101 SET character_set_client = utf8 */;
106 CREATE TABLE `class_schedule` (
107   `id` int(11) NOT NULL AUTO_INCREMENT,
108   `course_id` int(11) NOT NULL,
109   `subject_id` int(11) NOT NULL,
110   `teacher_id` int(11) NOT NULL,
111   `school_year_id` int(11) NOT NULL,
112   `day_of_week` enum('lunes','martes','miercoles','jueves','viernes') NOT NULL,
113   `start_time` time NOT NULL,
114   `end_time` time NOT NULL,
115   `period_number` int(11) DEFAULT NULL,
116   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
117   PRIMARY KEY (`id`),
118   KEY `course_id` (`course_id`),
119   KEY `subject_id` (`subject_id`),
120   KEY `teacher_id` (`teacher_id`),
121   KEY `school_year_id` (`school_year_id`),
122   CONSTRAINT `class_schedule_ibfk_1` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
123   CONSTRAINT `class_schedule_ibfk_2` FOREIGN KEY (`subject_id`) REFERENCES `subjects` (`id`),
124   CONSTRAINT `class_schedule_ibfk_3` FOREIGN KEY (`teacher_id`) REFERENCES `users` (`id`),
125   CONSTRAINT `class_schedule_ibfk_4` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`)
126 ) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
127 /*!40101 SET character_set_client = @saved_cs_client */;
128
129 --
130 -- Dumping data for table `class_schedule`
131 --
132
133 LOCK TABLES `class_schedule` WRITE;
134 /*!40000 ALTER TABLE `class_schedule` DISABLE KEYS */;
135 INSERT INTO `class_schedule` VALUES (2,5,3,5,1,'martes','00:00:00','00:00:00',7,'2026-02-15 03:05:28'),(5,5,6,7,1,'lunes...
136 /*!40000 ALTER TABLE `class_schedule` ENABLE KEYS */;
137 UNLOCK TABLES;
138
139 --
140 -- Table structure for table `course_students`
141 --
142
143 DROP TABLE IF EXISTS `course_students`;
144 /*!40101 SET @saved_cs_client = @@character_set_client */;
145 /*!40101 SET character_set_client = utf8 */;
146 CREATE TABLE `course_students` (
147   `id` int(11) NOT NULL AUTO_INCREMENT,
148   `student_id` int(11) NOT NULL,
149   `course_id` int(11) NOT NULL,
150   `school_year_id` int(11) NOT NULL,
151   `enrollment_date` date DEFAULT NULL,
152   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
153   PRIMARY KEY (`id`),
154   UNIQUE KEY `unique_enrollment` (`student_id`,`course_id`,`school_year_id`),

```

```

155     KEY `course_id` (`course_id`),
156     KEY `school_year_id` (`school_year_id`),
157     CONSTRAINT `course_students_ibfk_1` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),
158     CONSTRAINT `course_students_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
159     CONSTRAINT `course_students_ibfk_3` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`)
160 ) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

161 /*!40101 SET character_set_client = @saved_cs_client */;
162
163 --
164 -- Dumping data for table `course_students`
165 --
166
167 LOCK TABLES `course_students` WRITE;
168 /*!40000 ALTER TABLE `course_students` DISABLE KEYS */;
169 INSERT INTO `course_students` VALUES (1,8,1,1,'2025-09-01','2026-02-11 23:56:54'),(2,9,1,1,'2025-09-01','2026-02-11 23:5...
170 /*!40000 ALTER TABLE `course_students` ENABLE KEYS */;
171 UNLOCK TABLES;
172
173 --
174 -- Table structure for table `courses`
175 --
176
177 DROP TABLE IF EXISTS `courses`;
178 /*!40101 SET @saved_cs_client      = @@character_set_client */;
179 /*!40101 SET character_set_client = utf8 */;
180 CREATE TABLE `courses` (
181   `id` int(11) NOT NULL AUTO_INCREMENT,
182   `institution_id` int(11) NOT NULL,
183   `school_year_id` int(11) NOT NULL,
184   `name` varchar(100) NOT NULL,
185   `grade_level` varchar(50) DEFAULT NULL,
186   `parallel` varchar(10) DEFAULT NULL,
187   `shift_id` int(11) NOT NULL,
188   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
189   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
190   PRIMARY KEY (`id`),
191   KEY `institution_id` (`institution_id`),
192   KEY `school_year_id` (`school_year_id`),
193   KEY `shift_id` (`shift_id`),
194   CONSTRAINT `courses_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`),
195   CONSTRAINT `courses_ibfk_2` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`),
196   CONSTRAINT `courses_ibfk_3` FOREIGN KEY (`shift_id`) REFERENCES `shifts` (`id`)
197 ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
198 /*!40101 SET character_set_client = @saved_cs_client */;
199
200 --
201 -- Dumping data for table `courses`
202 --
203
204 LOCK TABLES `courses` WRITE;
205 /*!40000 ALTER TABLE `courses` DISABLE KEYS */;
206 INSERT INTO `courses` VALUES (1,1,1,'8vo EGB \"A\" - Matutina','8vo EGB','A',1,'2026-02-11 23:56:54','2026-02-17 22:08:4...
207 /*!40000 ALTER TABLE `courses` ENABLE KEYS */;
208 UNLOCK TABLES;
209
210 --
211 -- Table structure for table `institution_shifts`
212 --
213
214 DROP TABLE IF EXISTS `institution_shifts`;
215 /*!40101 SET @saved_cs_client      = @@character_set_client */;
216 /*!40101 SET character_set_client = utf8 */;
217 CREATE TABLE `institution_shifts` (
218   `id` int(11) NOT NULL AUTO_INCREMENT,
219   `institution_id` int(11) NOT NULL,
220   `shift_id` int(11) NOT NULL,
221   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
222   PRIMARY KEY (`id`),
223   UNIQUE KEY `unique_institution_shift` (`institution_id`,`shift_id`),
224   KEY `shift_id` (`shift_id`),
225   CONSTRAINT `institution_shifts_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`) ON DELETE CASCADE...
226   CONSTRAINT `institution_shifts_ibfk_2` FOREIGN KEY (`shift_id`) REFERENCES `shifts` (`id`) ON DELETE CASCADE
227 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
228 /*!40101 SET character_set_client = @saved_cs_client */;
229
230 --
231 -- Dumping data for table `institution_shifts`
232 --
233

```

```

234 LOCK TABLES `institution_shifts` WRITE;
235 /*!40000 ALTER TABLE `institution_shifts` DISABLE KEYS */;
236 INSERT INTO `institution_shifts` VALUES (1,1,1,'2026-02-15 11:41:49'),(2,1,2,'2026-02-15 11:41:49');
237 /*!40000 ALTER TABLE `institution_shifts` ENABLE KEYS */;
238 UNLOCK TABLES;
239
240 --
241
242 -- Table structure for table `institutions`
243 --
244 DROP TABLE IF EXISTS `institutions`;
245 /*!40101 SET @saved_cs_client      = @@character_set_client */;
246 /*!40101 SET character_set_client = utf8 */;
247 CREATE TABLE `institutions` (
248   `id` int(11) NOT NULL AUTO_INCREMENT,
249   `name` varchar(255) NOT NULL,
250   `code` varchar(50) NOT NULL,
251   `address` text DEFAULT NULL,
252   `province` varchar(100) DEFAULT NULL,
253   `city` varchar(100) DEFAULT NULL,
254   `phone` varchar(20) DEFAULT NULL,
255   `email` varchar(100) DEFAULT NULL,
256   `director_name` varchar(200) DEFAULT NULL,
257   `amie_code` varchar(20) DEFAULT NULL,
258   `website` varchar(255) DEFAULT NULL,
259   `logo_path` varchar(255) DEFAULT NULL,
260   `is_active` tinyint(1) DEFAULT 1,
261   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
262   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
263   PRIMARY KEY (`id`),
264   UNIQUE KEY `code` (`code`)
265 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
266 /*!40101 SET character_set_client = @saved_cs_client */;
267
268 --
269 -- Dumping data for table `institutions`
270 --
271
272 LOCK TABLES `institutions` WRITE;
273 /*!40000 ALTER TABLE `institutions` DISABLE KEYS */;
274 INSERT INTO `institutions` VALUES (1,'Unidad Educativa Pomasqui','UEP001','Av. Manuel Cordova Galarza 123, Quito, Ecuado...
275 /*!40000 ALTER TABLE `institutions` ENABLE KEYS */;
276 UNLOCK TABLES;
277
278 --
279 -- Table structure for table `justifications`
280 --
281
282 DROP TABLE IF EXISTS `justifications`;
283 /*!40101 SET @saved_cs_client      = @@character_set_client */;
284 /*!40101 SET character_set_client = utf8 */;
285 CREATE TABLE `justifications` (
286   `id` int(11) NOT NULL AUTO_INCREMENT,
287   `attendance_id` int(11) NOT NULL,
288   `student_id` int(11) NOT NULL,
289   `submitted_by` int(11) NOT NULL,
290   `reason` text NOT NULL,
291   `document_path` varchar(255) DEFAULT NULL,
292   `status` enum('pendiente','aprobado','rechazado') DEFAULT 'pendiente',
293   `reviewed_by` int(11) DEFAULT NULL,
294   `review_notes` text DEFAULT NULL,
295   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
296   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
297   PRIMARY KEY (`id`),
298   KEY `attendance_id` (`attendance_id`),
299   KEY `submitted_by` (`submitted_by`),
300   KEY `reviewed_by` (`reviewed_by`),
301   KEY `idx_status` (`status`),
302   KEY `idx_student` (`student_id`),
303   CONSTRAINT `justifications_ibfk_1` FOREIGN KEY (`attendance_id`) REFERENCES `attendances` (`id`) ON DELETE CASCADE,
304   CONSTRAINT `justifications_ibfk_2` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),
305   CONSTRAINT `justifications_ibfk_3` FOREIGN KEY (`submitted_by`) REFERENCES `users` (`id`),
306   CONSTRAINT `justifications_ibfk_4` FOREIGN KEY (`reviewed_by`) REFERENCES `users` (`id`)
307 ) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
308 /*!40101 SET character_set_client = @saved_cs_client */;
309
310 --
311 -- Dumping data for table `justifications`
312 --

```

```

313
314 LOCK TABLES `justifications` WRITE;
315 /*!40000 ALTER TABLE `justifications` DISABLE KEYS */;
316 INSERT INTO `justifications` VALUES (1,44,12,12,'Médico','uploads/justifications/69935b0811b11.png','rechazado',1,'dfasf...
317 /*!40000 ALTER TABLE `justifications` ENABLE KEYS */;
318 UNLOCK TABLES;
319
320 --
321 -- Table structure for table `notifications`
322 --
323
324 DROP TABLE IF EXISTS `notifications`;
325 /*!40101 SET @saved_cs_client      = @@character_set_client */;
326 /*!40101 SET character_set_client = utf8 */;
327 CREATE TABLE `notifications` (
328   `id` int(11) NOT NULL AUTO_INCREMENT,
329   `user_id` int(11) NOT NULL,
330   `title` varchar(255) NOT NULL,
331   `message` text NOT NULL,
332   `type` enum('info','warning','success','danger') DEFAULT 'info',
333   `link` varchar(255) DEFAULT NULL,
334   `is_read` tinyint(1) DEFAULT 0,
335   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
336   PRIMARY KEY (`id`),
337   KEY `idx_user_read` (`user_id`,`is_read`),
338   CONSTRAINT `notifications_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE
339 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
340 /*!40101 SET character_set_client = @saved_cs_client */;
341
342 --
343 -- Dumping data for table `notifications`
344 --
345
346 LOCK TABLES `notifications` WRITE;
347 /*!40000 ALTER TABLE `notifications` DISABLE KEYS */;
348 INSERT INTO `notifications` VALUES (1,1,'Nueva justificación pendiente','Un estudiante envió una justificación que requi...
349 /*!40000 ALTER TABLE `notifications` ENABLE KEYS */;
350 UNLOCK TABLES;
351
352 --
353 -- Table structure for table `permissions`
354 --
355
356 DROP TABLE IF EXISTS `permissions`;
357 /*!40101 SET @saved_cs_client      = @@character_set_client */;
358 /*!40101 SET character_set_client = utf8 */;
359 CREATE TABLE `permissions` (
360   `id` int(11) NOT NULL AUTO_INCREMENT,
361   `user_id` int(11) NOT NULL,
362   `permission_type` varchar(50) NOT NULL,
363   `entity_id` int(11) DEFAULT NULL,
364   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
365   PRIMARY KEY (`id`),
366   KEY `user_id` (`user_id`),
367   CONSTRAINT `permissions_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE
368 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
369 /*!40101 SET character_set_client = @saved_cs_client */;
370
371 --
372 -- Dumping data for table `permissions`
373 --
374
375 LOCK TABLES `permissions` WRITE;
376 /*!40000 ALTER TABLE `permissions` DISABLE KEYS */;
377 /*!40000 ALTER TABLE `permissions` ENABLE KEYS */;
378 UNLOCK TABLES;
379
380 --
381 -- Table structure for table `representatives`
382 --
383
384 DROP TABLE IF EXISTS `representatives`;
385 /*!40101 SET @saved_cs_client      = @@character_set_client */;
386 /*!40101 SET character_set_client = utf8 */;
387 CREATE TABLE `representatives` (
388   `id` int(11) NOT NULL AUTO_INCREMENT,
389   `representative_id` int(11) NOT NULL,
390   `student_id` int(11) NOT NULL,
391   `relationship` varchar(50) DEFAULT NULL,

```

```

392     `is_primary` tinyint(1) DEFAULT 0,
393     `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
394     PRIMARY KEY (`id`),
395     UNIQUE KEY `unique_rep_student` (`representative_id`,`student_id`),
396     KEY `student_id` (`student_id`),
397     CONSTRAINT `representatives_ibfk_1` FOREIGN KEY (`representative_id`) REFERENCES `users` (`id`),
398     CONSTRAINT `representatives_ibfk_2` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`)
399 ) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
400 /*!40101 SET character_set_client = @saved_cs_client */;

401
402 --
403 -- Dumping data for table `representatives`
404 --
405
406 LOCK TABLES `representatives` WRITE;
407 /*!40000 ALTER TABLE `representatives` DISABLE KEYS */;
408 INSERT INTO `representatives` VALUES (1,23,12,'Abuelo/a',0,'2026-02-11 23:56:54'),(2,23,13,'Madre',1,'2026-02-11 23:56:5...
409 /*!40000 ALTER TABLE `representatives` ENABLE KEYS */;
410 UNLOCK TABLES;
411
412 --
413 -- Table structure for table `roles`
414 --
415
416 DROP TABLE IF EXISTS `roles`;
417 /*!40101 SET @saved_cs_client      = @@character_set_client */;
418 /*!40101 SET character_set_client = utf8 */;
419 CREATE TABLE `roles` (
420     `id` int(11) NOT NULL AUTO_INCREMENT,
421     `name` varchar(50) NOT NULL,
422     `description` text DEFAULT NULL,
423     `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
424     `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
425     PRIMARY KEY (`id`),
426     UNIQUE KEY `name` (`name`)
427 ) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
428 /*!40101 SET character_set_client = @saved_cs_client */;
429
430 --
431 -- Dumping data for table `roles`
432 --
433
434 LOCK TABLES `roles` WRITE;
435 /*!40000 ALTER TABLE `roles` DISABLE KEYS */;
436 INSERT INTO `roles` VALUES (1,'docente','Profesor de la institución','2026-02-11 23:56:54','2026-02-11 23:56:54'),(2,'es...
437 /*!40000 ALTER TABLE `roles` ENABLE KEYS */;
438 UNLOCK TABLES;
439
440 --
441 -- Table structure for table `school_years`
442 --
443
444 DROP TABLE IF EXISTS `school_years`;
445 /*!40101 SET @saved_cs_client      = @@character_set_client */;
446 /*!40101 SET character_set_client = utf8 */;
447 CREATE TABLE `school_years` (
448     `id` int(11) NOT NULL AUTO_INCREMENT,
449     `institution_id` int(11) NOT NULL,
450     `name` varchar(100) NOT NULL,
451     `start_date` date NOT NULL,
452     `end_date` date NOT NULL,
453     `is_active` tinyint(1) DEFAULT 0,
454     `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
455     `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
456     PRIMARY KEY (`id`),
457     KEY `institution_id` (`institution_id`),
458     CONSTRAINT `school_years_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`)
459 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
460 /*!40101 SET character_set_client = @saved_cs_client */;
461
462 --
463 -- Dumping data for table `school_years`
464 --
465
466 LOCK TABLES `school_years` WRITE;
467 /*!40000 ALTER TABLE `school_years` DISABLE KEYS */;
468 INSERT INTO `school_years` VALUES (1,1,'2025-2026','2025-09-01','2026-07-30',1,'2026-02-11 23:56:54','2026-02-11 23:56:5...
469 /*!40000 ALTER TABLE `school_years` ENABLE KEYS */;
470 UNLOCK TABLES;

```

```

471
472 --
473 -- Table structure for table `shifts`
474 --
475
476 DROP TABLE IF EXISTS `shifts`;
477 /*!40101 SET @saved_cs_client      = @@character_set_client */;
478 /*!40101 SET character_set_client = utf8 */;
479 CREATE TABLE `shifts` (
480   `id` int(11) NOT NULL AUTO_INCREMENT,
481
482   `name` varchar(50) NOT NULL,
483   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
484   PRIMARY KEY (`id`)
485 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
486 /*!40101 SET character_set_client = @saved_cs_client */;
487 --
488 -- Dumping data for table `shifts`
489 --
490
491 LOCK TABLES `shifts` WRITE;
492 /*!40000 ALTER TABLE `shifts` DISABLE KEYS */;
493 INSERT INTO `shifts` VALUES (1,'matutina','2026-02-11 23:56:54'),(2,'vespertina','2026-02-11 23:56:54'),(3,'nocturna','2...
494 /*!40000 ALTER TABLE `shifts` ENABLE KEYS */;
495 UNLOCK TABLES;
496
497 --
498 -- Table structure for table `subjects`
499 --
500
501 DROP TABLE IF EXISTS `subjects`;
502 /*!40101 SET @saved_cs_client      = @@character_set_client */;
503 /*!40101 SET character_set_client = utf8 */;
504 CREATE TABLE `subjects` (
505   `id` int(11) NOT NULL AUTO_INCREMENT,
506   `institution_id` int(11) NOT NULL,
507   `name` varchar(100) NOT NULL,
508   `code` varchar(20) DEFAULT NULL,
509   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
510   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
511   PRIMARY KEY (`id`),
512   KEY `institution_id` (`institution_id`),
513   CONSTRAINT `subjects_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`)
514 ) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
515 /*!40101 SET character_set_client = @saved_cs_client */;
516
517 --
518 -- Dumping data for table `subjects`
519 --
520
521 LOCK TABLES `subjects` WRITE;
522 /*!40000 ALTER TABLE `subjects` DISABLE KEYS */;
523 INSERT INTO `subjects` VALUES (1,1,'Matemáticas','MAT','2026-02-11 23:56:54','2026-02-11 23:56:54'),(2,1,'Lengua y Liter...
524 /*!40000 ALTER TABLE `subjects` ENABLE KEYS */;
525 UNLOCK TABLES;
526
527 --
528 -- Table structure for table `teacher_assignments`
529 --
530
531 DROP TABLE IF EXISTS `teacher_assignments`;
532 /*!40101 SET @saved_cs_client      = @@character_set_client */;
533 /*!40101 SET character_set_client = utf8 */;
534 CREATE TABLE `teacher_assignments` (
535   `id` int(11) NOT NULL AUTO_INCREMENT,
536   `teacher_id` int(11) NOT NULL,
537   `course_id` int(11) NOT NULL,
538   `subject_id` int(11) NOT NULL,
539   `school_year_id` int(11) NOT NULL,
540   `is_tutor` tinyint(1) DEFAULT 0,
541   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
542   PRIMARY KEY (`id`),
543   KEY `teacher_id` (`teacher_id`),
544   KEY `course_id` (`course_id`),
545   KEY `subject_id` (`subject_id`),
546   KEY `school_year_id` (`school_year_id`),
547   CONSTRAINT `teacher_assignments_ibfk_1` FOREIGN KEY (`teacher_id`) REFERENCES `users` (`id`),
548   CONSTRAINT `teacher_assignments_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
549   CONSTRAINT `teacher_assignments_ibfk_3` FOREIGN KEY (`subject_id`) REFERENCES `subjects` (`id`),

```



```

550     CONSTRAINT `teacher_assignments_ibfk_4` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`)
551 ) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
552 /*!40101 SET character_set_client = @saved_cs_client */;
553
554 --
555 -- Dumping data for table `teacher_assignments`
556 --
557
558 LOCK TABLES `teacher_assignments` WRITE;
559 /*!40000 ALTER TABLE `teacher_assignments` DISABLE KEYS */;
560 INSERT INTO `teacher_assignments` VALUES (1,2,1,1,1,0,'2026-02-11 23:56:54'),(2,2,2,1,1,1,'2026-02-11 23:56:54'),(3,2,3,...

561 /*!40000 ALTER TABLE `teacher_assignments` ENABLE KEYS */;
562 UNLOCK TABLES;
563
564 --
565 -- Table structure for table `user_roles`
566 --
567
568 DROP TABLE IF EXISTS `user_roles`;
569 /*!40101 SET @saved_cs_client      = @@character_set_client */;
570 /*!40101 SET character_set_client = utf8 */;
571 CREATE TABLE `user_roles` (
572   `id` int(11) NOT NULL AUTO_INCREMENT,
573   `user_id` int(11) NOT NULL,
574   `role_id` int(11) NOT NULL,
575   `assigned_at` timestamp NOT NULL DEFAULT current_timestamp(),
576   PRIMARY KEY (`id`),
577   UNIQUE KEY `unique_user_role` (`user_id`,`role_id`),
578   KEY `role_id` (`role_id`),
579   CONSTRAINT `user_roles_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,
580   CONSTRAINT `user_roles_ibfk_2` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`) ON DELETE CASCADE
581 ) ENGINE=InnoDB AUTO_INCREMENT=43 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
582 /*!40101 SET character_set_client = @saved_cs_client */;
583
584 --
585 -- Dumping data for table `user_roles`
586 --
587
588 LOCK TABLES `user_roles` WRITE;
589 /*!40000 ALTER TABLE `user_roles` DISABLE KEYS */;
590 INSERT INTO `user_roles` VALUES (2,2,1,'2026-02-11 23:56:54'),(3,3,1,'2026-02-11 23:56:54'),(4,4,1,'2026-02-11 23:56:54'...
591 /*!40000 ALTER TABLE `user_roles` ENABLE KEYS */;
592 UNLOCK TABLES;
593
594 --
595 -- Table structure for table `users`
596 --
597
598 DROP TABLE IF EXISTS `users`;
599 /*!40101 SET @saved_cs_client      = @@character_set_client */;
600 /*!40101 SET character_set_client = utf8 */;
601 CREATE TABLE `users` (
602   `id` int(11) NOT NULL AUTO_INCREMENT,
603   `institution_id` int(11) NOT NULL,
604   `username` varchar(50) NOT NULL,
605   `email` varchar(100) NOT NULL,
606   `password` varchar(255) NOT NULL,
607   `first_name` varchar(100) NOT NULL,
608   `last_name` varchar(100) NOT NULL,
609   `dni` varchar(20) DEFAULT NULL,
610   `phone` varchar(20) DEFAULT NULL,
611   `reset_token` varchar(255) DEFAULT NULL,
612   `reset_expires` datetime DEFAULT NULL,
613   `is_active` tinyint(1) DEFAULT 1,
614   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
615   `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
616   PRIMARY KEY (`id`),
617   UNIQUE KEY `username` (`username`),
618   UNIQUE KEY `email` (`email`),
619   UNIQUE KEY `dni` (`dni`),
620   KEY `institution_id` (`institution_id`),
621   CONSTRAINT `users_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`)
622 ) ENGINE=InnoDB AUTO_INCREMENT=32 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
623 /*!40101 SET character_set_client = @saved_cs_client */;
624
625 --
626 -- Dumping data for table `users`
627 --
628

```

```
629 LOCK TABLES `users` WRITE;
630 /*!40000 ALTER TABLE `users` DISABLE KEYS */;
631 INSERT INTO `users` VALUES (1,1,'admin','admin@uep.edu.ec','$2y$10$92IXUNpkj00rQQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi...
632 /*!40000 ALTER TABLE `users` ENABLE KEYS */;
633 UNLOCK TABLES;
634 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
635
636 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
637 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
638 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
639 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
640 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

641 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
642 /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
643
644 -- Dump completed on 2026-02-17 23:49:44
645
```

### 3. config.php

Ruta: ArchImp/config/config.php

```
1  <?php
2  // Zona horaria Ecuador
3  date_default_timezone_set('America/Guayaquil');
4
5  // Definir constantes ANTES de todo
6  define('BASE_PATH', __DIR__ . '/../');
7  define('BASE_URL', 'http://localhost/ecuasistencia2026');
8  define('EDIT_ATTENDANCE_HOURS', 48);
9
10 // Configuración de sesión ANTES de session_start()
11 ini_set('session.gc_maxlifetime', 86400); // 24 horas
12 session_set_cookie_params([
13     'lifetime' => 86400, // 24 horas
14     'path' => '/',
15     'domain' => '',
16     'secure' => false, // true si usas HTTPS
17     'httponly' => true, // Protección XSS
18     'samesite' => 'Lax' // Protección CSRF
19 ]);
20
21 // Iniciar sesión DESPUÉS de configurar
22 session_start();
23
24 // Timeout de inactividad: 30 minutos
25 $inactive_timeout = 1800; // 30 minutos en segundos
26 if (isset($_SESSION['last_activity']) &&
27     (time() - $_SESSION['last_activity'] > $inactive_timeout)) {
28     session_unset();
29     session_destroy();
30     header('Location: ' . BASE_URL . '/public/?action=login&timeout=1');
31     exit;
32 }
33 $_SESSION['last_activity'] = time();
34
35 // SMTP Config
36 define('SMTP_HOST', 'smtp.gmail.com');
37 define('SMTP_PORT', 587);
38 define('SMTP_USER', 'viktorengel@gmail.com');
39 define('SMTP_PASS', 'Orktvi.5/*83gM');
40 define('SMTP_FROM', 'noreply@ecuasist.edu.ec');
41 define('SMTP_NAME', 'EcuAsist2026');
42
43 require_once BASE_PATH . '/config/database.php';
44 require_once BASE_PATH . '/helpers/Security.php';
```

## 4. database.php

Ruta: 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 }
```

## 5. AcademicController.php

Ruta: 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
16     public function __construct() {
17         Security::requireLogin();
18         if (!Security::hasRole('autoridad')) {
19             die('Acceso denegado');
20         }
21
22         $db = new Database();
23         $this->courseModel = new Course($db);
24         $this->subjectModel = new Subject($db);
25         $this->schoolYearModel = new SchoolYear($db);
26         $this->shiftModel = new Shift($db);
27         $this->userModel = new User($db);
28     }
29
30     public function index() {
31         $courses = $this->courseModel->getAll();
32         $subjects = $this->subjectModel->getAll();
33         $schoolYears = $this->schoolYearModel->getAll();
34         $shifts = $this->shiftModel->getAll();
35
36         include BASE_PATH . '/views/academic/index.php';
37     }
38
39     public function createCourse() {
40         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
41             $activeYear = $this->schoolYearModel->getActive();
42
43             if (!$activeYear) {
44                 header('Location: ?action=academic&error=no_active_year');
45                 exit;
46             }
47
48             $data = [
49                 'institution_id' => $_SESSION['institution_id'],
50                 'school_year_id' => $activeYear['id'],
51                 'name' => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
52                 'grade_level' => html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8'),
53                 'parallel' => Security::sanitize($_POST['parallel']),
54                 'shift_id' => (int)$_POST['shift_id']
55             ];
56
57             $this->courseModel->create($data);
58             header('Location: ?action=academic&course_success=1');
59             exit;
60         }
61     }
62
63     public function createSubject() {
64         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
65             $data = [
66                 'institution_id' => $_SESSION['institution_id'],
67                 'name' => Security::sanitize($_POST['name']),
68                 'code' => Security::sanitize($_POST['code'])
69             ];
70
71             $this->subjectModel->create($data);
72             header('Location: ?action=academic&subject_success=1');
73             exit;
74         }
75     }
76 }
```

```

76
77 // =====
78 // CRUD CURSOS
79 // =====
80
81 public function editCourse() {
82     $courseId = (int)$GET['id'];
83     $course = $this->courseModel->findById($courseId);
84
85     if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
86         header('Location: ?action=academic&error=course_not_found');
87         exit;
88     }
89
90     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
91         $shifts = $this->shiftModel->getAll();
92         include BASE_PATH . '/views/academic/course_edit.php';
93         return;
94     }
95
96     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
97         $data = [
98             'id' => $courseId,
99             'name' => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
100             'grade_level' => html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8'),
101             'parallel' => Security::sanitize($_POST['parallel']),
102             'shift_id' => (int)$_POST['shift_id']
103         ];
104
105         if ($this->courseModel->update($data)) {
106             header('Location: ?action=academic&course_updated=1');
107             exit;
108         } else {
109             $errors[] = "Error al actualizar el curso";
110             $shifts = $this->shiftModel->getAll();
111             include BASE_PATH . '/views/academic/course_edit.php';
112         }
113     }
114 }
115
116 public function deleteCourse() {
117     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
118         $courseId = (int)$_POST['course_id'];
119         $course = $this->courseModel->findById($courseId);
120
121         if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
122             header('Location: ?action=academic&error=course_not_found');
123             exit;
124         }
125
126         // Verificar si tiene estudiantes matriculados
127         $students = $this->courseModel->getEnrolledStudents($courseId);
128         if (count($students) > 0) {
129             header('Location: ?action=academic&error=course_has_students');
130             exit;
131         }
132
133         // Verificar si tiene asignaciones docentes
134         $db = new Database();
135         $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM teacher_assignments WHERE course_id = ...");
136         $stmt->execute([':id' => $courseId]);
137         $result = $stmt->fetch();
138
139         if ($result['count'] > 0) {
140             header('Location: ?action=academic&error=course_has_assignments');
141             exit;
142         }
143
144         if ($this->courseModel->delete($courseId)) {
145             header('Location: ?action=academic&course_deleted=1');
146             exit;
147         } else {
148             header('Location: ?action=academic&error=delete_failed');
149             exit;
150         }
151     }
152 }
153
154 // =====

```

```

155 // CRUD ASIGNATURAS
156 // =====
157
158 public function editSubject() {
159     $subjectId = (int)$GET['id'];
160     $subject = $this->subjectModel->findById($subjectId);
161
162     if (!$subject || $subject['institution_id'] != $_SESSION['institution_id']) {
163         header('Location: ?action=academic&error=subject_not_found');
164         exit;
165     }
166
167     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
168         include BASE_PATH . '/views/academic/subject_edit.php';
169         return;
170     }
171
172     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
173         $data = [
174             'id' => $subjectId,
175             'name' => Security::sanitize($_POST['name']),
176             'code' => Security::sanitize($_POST['code'])
177         ];
178
179         if ($this->subjectModel->update($data)) {
180             header('Location: ?action=academic&subject_updated=1');
181             exit;
182         } else {
183             $errors[] = "Error al actualizar la asignatura";
184             include BASE_PATH . '/views/academic/subject_edit.php';
185         }
186     }
187 }
188
189 public function deleteSubject() {
190     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
191         $subjectId = (int)$_POST['subject_id'];
192         $subject = $this->subjectModel->findById($subjectId);
193
194         if (!$subject || $subject['institution_id'] != $_SESSION['institution_id']) {
195             header('Location: ?action=academic&error=subject_not_found');
196             exit;
197         }
198
199         // Verificar si tiene asignaciones docentes
200         $db = new Database();
201         $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM teacher_assignments WHERE subject_id = ...");
202         $stmt->execute([':id' => $subjectId]);
203         $result = $stmt->fetch();
204
205         if ($result['count'] > 0) {
206             header('Location: ?action=academic&error=subject_has_assignments');
207             exit;
208         }
209
210         if ($this->subjectModel->delete($subjectId)) {
211             header('Location: ?action=academic&subject_deleted=1');
212             exit;
213         } else {
214             header('Location: ?action=academic&error=delete_failed');
215             exit;
216         }
217     }
218 }
219
220 public function enrollStudents() {
221     $activeYear = $this->schoolYearModel->getActive();
222     $courses = $this->courseModel->getAll();
223     $availableStudents = $this->userModel->getStudentsNotEnrolled($activeYear['id']);
224     $allStudents = $this->userModel->getByRole('estudiante');
225
226     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
227         $courseId = (int)$_POST['course_id'];
228         $studentIds = $_POST['student_ids'] ?? [];
229
230         $enrolled = 0;
231         $errors = 0;
232
233         foreach ($studentIds as $studentId) {

```

```

234         if ($this->courseModel->enrollStudent($courseId, (int)$studentId, $activeYear['id'])) {
235             $enrolled++;
236         } else {
237             $errors++;
238         }
239     }
240
241     header('Location: ?action=enroll_students&enrolled=' . $enrolled . '&errors=' . $errors);
242     exit;
243 }
244
245 include BASE_PATH . '/views/academic/enroll.php';
246 }
247
248 public function unenrollStudent() {
249     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
250         $studentId = (int)$_POST['student_id'];
251         $activeYear = $this->schoolYearModel->getActive();
252
253         if (!$activeYear) {
254             header('Location: ?action=enroll_students&error=no_active_year');
255             exit;
256         }
257
258         // Verificar que el estudiante existe y está matriculado
259         $course = $this->userModel->getStudentCourse($studentId, $activeYear['id']);
260
261         if (!$course) {
262             header('Location: ?action=enroll_students&error=not_enrolled');
263             exit;
264         }
265
266         // Retirar estudiante
267         if ($this->courseModel->unenrollStudent($studentId, $activeYear['id'])) {
268             header('Location: ?action=enroll_students&unenrolled=1');
269             exit;
270         } else {
271             header('Location: ?action=enroll_students&error=unenroll_failed');
272             exit;
273         }
274     }
275 }
276
277 public function viewCourseStudents() {
278     $courseId = (int)($_GET['course_id'] ?? 0);
279
280     if (!$courseId) {
281         header('Location: ?action=academic');
282         exit;
283     }
284
285     $course = $this->courseModel->getAll();
286     $course = array_filter($course, fn($c) => $c['id'] == $courseId);
287     $course = reset($course);
288
289     $students = $this->courseModel->getEnrolledStudents($courseId);
290
291     include BASE_PATH . '/views/academic/course_students.php';
292 }
293
294 // =====
295 // CRUD AÑOS LECTIVOS
296 // =====
297
298 public function createSchoolYear() {
299     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
300         include BASE_PATH . '/views/academic/school_year_create.php';
301         return;
302     }
303
304     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
305         $errors = [];
306
307         // Validaciones
308         if (empty($_POST['name'])) {
309             $errors[] = "El nombre es obligatorio";
310         }
311         if (empty($_POST['start_date'])) {
312             $errors[] = "La fecha de inicio es obligatoria";

```



```

313     }
314     if (empty($_POST['end_date'])) {
315         $errors[] = "La fecha de fin es obligatoria";
316     }
317
318     if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {
319         if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
320             $errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
321         }
322
323         // Verificar solapamiento de fechas
324         if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'])) {
325             $errors[] = "Ya existe un año lectivo con fechas que se solapan";
326         }
327     }
328
329     if (!empty($errors)) {
330         include BASE_PATH . '/views/academic/school_year_create.php';
331         return;
332     }
333
334     // Crear año lectivo
335     $data = [
336         'institution_id' => $_SESSION['institution_id'],
337         'name' => Security::sanitize($_POST['name']),
338         'start_date' => $_POST['start_date'],
339         'end_date' => $_POST['end_date'],
340         'is_active' => isset($_POST['is_active']) ? 1 : 0
341     ];
342
343     // Si se marca como activo, desactivar los demás
344     if ($data['is_active']) {
345         $this->schoolYearModel->activate(0); // Desactiva todos
346     }
347
348     if ($this->schoolYearModel->create($data)) {
349         // Si se creó como activo, activar el nuevo
350         if ($data['is_active']) {
351             $db = new Database();
352             $newId = $db->connect()->lastInsertId();
353             $this->schoolYearModel->activate($newId);
354         }
355
356         header('Location: ?action=academic&sy_created=1');
357         exit;
358     } else {
359         $errors[] = "Error al crear el año lectivo";
360         include BASE_PATH . '/views/academic/school_year_create.php';
361     }
362 }
363 }
364
365 public function editSchoolYear() {
366     $yearId = (int)$_GET['id'];
367     $year = $this->schoolYearModel->findById($yearId);
368
369     if (!$year) {
370         header('Location: ?action=academic&error=year_not_found');
371         exit;
372     }
373
374     // Verificar institución
375     if ($year['institution_id'] != $_SESSION['institution_id']) {
376         die('Acceso denegado');
377     }
378
379     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
380         include BASE_PATH . '/views/academic/school_year_edit.php';
381         return;
382     }
383
384     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
385         $errors = [];
386
387         // Validaciones
388         if (empty($_POST['name'])) {
389             $errors[] = "El nombre es obligatorio";
390         }
391         if (empty($_POST['start_date'])) {

```

```

392         $errors[] = "La fecha de inicio es obligatoria";
393     }
394     if (empty($_POST['end_date'])) {
395         $errors[] = "La fecha de fin es obligatoria";
396     }
397
398     if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {
399         if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
400             $errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
401         }
402
403         // Verificar solapamiento (excluyendo el actual)
404         if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'], $yearId)) {
405             $errors[] = "Ya existe otro año lectivo con fechas que se solapan";
406         }
407     }
408
409     if (!empty($errors)) {
410         include BASE_PATH . '/views/academic/school_year_edit.php';
411         return;
412     }
413
414     // Actualizar año lectivo
415     $data = [
416         'id' => $yearId,
417         'name' => Security::sanitize($_POST['name']),
418         'start_date' => $_POST['start_date'],
419         'end_date' => $_POST['end_date']
420     ];
421
422     if ($this->schoolYearModel->update($data)) {
423         header('Location: ?action=academic&sy_updated=1');
424         exit;
425     } else {
426         $errors[] = "Error al actualizar el año lectivo";
427         include BASE_PATH . '/views/academic/school_year_edit.php';
428     }
429 }
430 }
431
432 public function deleteSchoolYear() {
433     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
434         $yearId = (int)$_POST['year_id'];
435         $year = $this->schoolYearModel->findById($yearId);
436
437         if (!$year) {
438             header('Location: ?action=academic&error=year_not_found');
439             exit;
440         }
441
442         // Verificar institución
443         if ($year['institution_id'] != $_SESSION['institution_id']) {
444             die('Acceso denegado');
445         }
446
447         // No permitir eliminar año activo
448         if ($year['is_active'] == 1) {
449             header('Location: ?action=academic&error=cannot_delete_active');
450             exit;
451         }
452
453         if ($this->schoolYearModel->delete($yearId)) {
454             header('Location: ?action=academic&sy_deleted=1');
455             exit;
456         } else {
457             header('Location: ?action=academic&error=has_courses');
458             exit;
459         }
460     }
461 }
462
463 public function activateSchoolYear() {
464     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
465         $yearId = (int)$_POST['year_id'];
466         $year = $this->schoolYearModel->findById($yearId);
467
468         if (!$year) {
469             header('Location: ?action=academic&error=year_not_found');
470             exit;

```

```

471     }
472
473     // Verificar institución
474     if ($year['institution_id'] != $_SESSION['institution_id']) {
475         die('Acceso denegado');
476     }
477
478     if ($this->schoolYearModel->activate($yearId)) {
479         header('Location: ?action=academic&sy_activated=1');
480         exit;
481
482     } else {
483         header('Location: ?action=academic&error=activate_failed');
484         exit;
485     }
486 }
487
488 public function deactivateSchoolYear() {
489     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
490         $yearId = (int)$_POST['year_id'];
491         $year = $this->schoolYearModel->findById($yearId);
492
493         if (!$year) {
494             header('Location: ?action=academic&error=year_not_found');
495             exit;
496         }
497
498         // Verificar institución
499         if ($year['institution_id'] != $_SESSION['institution_id']) {
500             die('Acceso denegado');
501         }
502
503         if ($this->schoolYearModel->deactivate($yearId)) {
504             header('Location: ?action=academic&sy_deactivated=1');
505             exit;
506         } else {
507             header('Location: ?action=academic&error=deactivate_failed');
508             exit;
509         }
510     }
511 }
512 }

```

## 6. AssignmentController.php

Ruta: 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, $activeYear['id']);
156
157     include BASE_PATH . '/views/assignments/view_course.php';
158 }
159
160 public function getCourseTeachers() {
161
162     if (!isset($_SESSION['user_id'])) {
163         header('Content-Type: application/json');
164         echo json_encode(['error' => 'No autenticado']);
165         exit;
166     }
167
168     $courseId = (int)($_GET['course_id'] ?? 0);
169
170     if (!$courseId) {
171         header('Content-Type: application/json');
172         echo json_encode([]);
173         exit;
174     }
175
176     // Obtener año lectivo activo
177     $db = new Database();
178     $sqlYear = "SELECT id FROM school_years WHERE is_active = 1 LIMIT 1";
179     $stmtYear = $db->connect()->query($sqlYear);
180     $activeYear = $stmtYear->fetch();
181
182     if (!$activeYear) {
183         header('Content-Type: application/json');
184         echo json_encode([]);
185         exit;
186     }
187
188     // Solo docentes del curso que:
189     // 1. NO son tutores de otro curso
190     // 2. NO son tutores del curso actual
191     $sql = "SELECT DISTINCT ta.teacher_id,
192             CONCAT(u.last_name, ' ', u.first_name) as teacher_name
193             FROM teacher_assignments ta
194             INNER JOIN users u ON ta.teacher_id = u.id
195             WHERE ta.course_id = :course_id
196             AND ta.teacher_id NOT IN (
197                 SELECT teacher_id
198                 FROM teacher_assignments
199                 WHERE is_tutor = 1
200                 AND school_year_id = :school_year_id
201             )
202             ORDER BY u.last_name, u.first_name";
203
204     $stmt = $db->connect()->prepare($sql);
205     $stmt->execute([
206         ':course_id' => $courseId,
207         ':school_year_id' => $activeYear['id']
208     ]);
209     $teachers = $stmt->fetchAll(PDO::FETCH_ASSOC);
210
211     header('Content-Type: application/json');
212     echo json_encode($teachers);
213     exit;
214 }
215
216 public function tutorManagement() {
217     $courses = $this->courseModel->getAll();
218     $assignments = $this->assignmentModel->getAll();
219
220     include BASE_PATH . '/views/assignments/tutor.php';
221 }
222
223 public function checkCourseTutor() {
224     if (!isset($_SESSION['user_id'])) {
225         header('Content-Type: application/json');
226         echo json_encode(null);
227         exit;
228     }
229
230     $courseId = (int)($_GET['course_id'] ?? 0);
231
232     if (!$courseId) {
233         header('Content-Type: application/json');
234         echo json_encode(null);
235     }

```

```

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 }

```

## 7. AttendanceController.php

Ruta: 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
82         $activeYear = $this->schoolYearModel->getActive();
83         $date = $_POST['date'];
84         $today = date('Y-m-d');
85
86         if (strtotime($date) > strtotime($today)) {
87             header('Location: ?action=attendance_register&error=future');
88             exit;
89         }
90
91         if (!$this->isWithin48BusinessHours($date)) {
92             header('Location: ?action=attendance_register&error=toolate');
93             exit;
94         }
95
96         $scheduleId = (int)$_POST['schedule_id'];
97
98         // Obtener datos del horario incluyendo nombre de asignatura
99         $db = new Database();
100        $pdo = $db->connect();
101        $sql = "SELECT cs.*, s.name as subject_name
102                FROM class_schedule cs
103                INNER JOIN subjects s ON cs.subject_id = s.id
104                WHERE cs.id = :id";
105        $stmt = $pdo->prepare($sql);
106        $stmt->execute([':id' => $scheduleId]);
107        $scheduleData = $stmt->fetch();
108
109        $subjectName = $scheduleData['subject_name'] ?? 'clase';
110        $dateFormatted = date('d/m/Y', strtotime($date));
111
112        foreach ($_POST as $key => $value) {
113            if (strpos($key, 'status_') === 0) {
114                $studentId = (int)str_replace('status_', '', $key);
115                $status = Security::sanitize($value);
116                $observation = Security::sanitize($_POST['obs_' . $studentId] ?? '');
117
118                $data = [
119                    ':student_id' => $studentId,
120                    ':course_id' => $scheduleData['course_id'],
121                    ':subject_id' => $scheduleData['subject_id'],
122                    ':teacher_id' => $_SESSION['user_id'],
123                    ':school_year_id' => $activeYear['id'],
124                    ':shift_id' => 1,
125                    ':date' => $date,
126                    ':hour_period' => $scheduleData['period_number'] . 'ra hora',
127                    ':status' => $status,
128                    ':observation' => $observation
129                ];
130
131                $this->attendanceModel->create($data);
132
133                // Notificar al estudiante si fue marcado ausente
134                // El enlace lleva directo a su asistencia para que pueda justificar
135                if ($status === 'ausente') {
136                    $this->notificationModel->create(
137                        $studentId,
138                        '■ Ausencia registrada',
139                        "Se registró una ausencia el {$dateFormatted} en {$subjectName}. Puedes justificarla.",
140                        'ausente',
141                        '?action=my_attendance'
142                    );
143                }
144
145                // Notificar al estudiante si fue marcado con tardanza
146                if ($status === 'tardanza') {
147                    $this->notificationModel->create(
148                        $studentId,
149                        '■ Tardanza registrada',
150                        "Se registró una tardanza el {$dateFormatted} en {$subjectName}.",
151                        'warning',
152                        '?action=my_attendance'
153                    );
154                }
155            }
156        }
157    }

```

```

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-&gt;execute([':id' => $scheduleId]);
393     $schedule = $stmt-&gt;fetch(PDO::FETCH_ASSOC);
394
395     header('Content-Type: application/json');
396     echo json_encode($schedule);
397     exit;
398 }
399
400 public function getExistingAttendance() {
401
402     header('Content-Type: application/json');
403
404     if (!isset($_SESSION['user_id'])) {
405         echo json_encode([]);
406         exit;
407     }
408
409     $scheduleId = (int)($_GET['schedule_id'] ?? 0);
410     $date        = $_GET['date'] ?? date('Y-m-d');
411
412     if (!$scheduleId) {
413         echo json_encode([]);
414         exit;
415     }
416
417     $db    = new Database();
418     $pdo   = $db-&gt;connect();
419     $sql    = "SELECT * FROM class_schedule WHERE id = :id";
420     $stmt   = $pdo-&gt;prepare($sql);
421     $stmt-&gt;execute([':id' => $scheduleId]);
422     $schedule = $stmt-&gt;fetch();
423
424     if (!$schedule) {
425         echo json_encode([]);
426         exit;
427     }
428
429     $sql2 = "SELECT student_id, status, observation
430             FROM attendances
431             WHERE course_id = :course_id
432             AND subject_id = :subject_id
433             AND date        = :date
434             AND hour_period = :hour_period";
435     $stmt2 = $pdo-&gt;prepare($sql2);
436     $stmt2-&gt;execute([
437         ':course_id' => $schedule['course_id'],
438         ':subject_id' => $schedule['subject_id'],
439         ':date'       => $date,
440         ':hour_period' => $schedule['period_number'] . 'ra hora'
441     ]);
442
443     echo json_encode($stmt2-&gt;fetchAll(PDO::FETCH_ASSOC));
444     exit;
445 }

```

## 8. AuthController.php

Ruta: 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
60                 header('Location: ' . BASE_URL . '/public/index.php?action=dashboard');
61                 exit;
62             } else {
63                 $error = 'Credenciales incorrectas';
64             }
65         }
66
67         include BASE_PATH . '/views/auth/login.php';
68     }
69
70     public function logout() {
71         session_destroy();
72         header('Location: ' . BASE_URL . '/public/index.php?action=login');
73         exit;
74     }
75 }
```

```

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

```

## 9. BackupController.php

Ruta: 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 }
```



## 10. DashboardController.php

Ruta: 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
162     return $row ? $row['course_name'] : null;
163 }
```

## 11. ExportController.php

Ruta: 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()
58                 ->setFillType(Fill::FILL_SOLID)
59                 ->getStartColor()->setRGB('007bff');
60             $sheet->getStyle($col . '6')->getFont()->getColor()->setRGB('FFFFFF');
61             $col++;
62         }
63
64         // Datos
65         $row = 7;
66         $num = 1;
67         foreach ($students as $student) {
68             $sheet->setCellValue('A' . $row, $num++);
69             $sheet->setCellValue('B' . $row, $student['last_name']);
70             $sheet->setCellValue('C' . $row, $student['first_name']);
71             $sheet->setCellValue('D' . $row, $student['dni'] ?? '-');
72             $sheet->setCellValue('E' . $row, date('d/m/Y', strtotime($student['enrollment_date'])));
73             $row++;
74         }
75     }
76 }
```

```
76 // Ajustar anchos
77 $sheet->getColumnDimension('A')->setWidth(5);
78 $sheet->getColumnDimension('B')->setWidth(20);
79 $sheet->getColumnDimension('C')->setWidth(20);
80 $sheet->getColumnDimension('D')->setWidth(15);

81 $sheet->getColumnDimension('E')->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->save('php://output');
90 exit;
91 }
92 }
```

## 12. InstitutionController.php

Ruta: 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->getId(1); // Por ahora solo una institución
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') {
34             $logoPath = null;
35
36             // Subir logo si existe
37             if (isset($_FILES['logo']) && $_FILES['logo']['error'] == 0) {
38                 $uploadDir = BASE_PATH . '/uploads/institution/';
39                 if (!file_exists($uploadDir)) {
40                     mkdir($uploadDir, 0755, true);
41                 }
42
43                 $extension = pathinfo($_FILES['logo']['name'], PATHINFO_EXTENSION);
44                 $filename = 'logo_' . time() . '.' . $extension;
45                 $uploadPath = $uploadDir . $filename;
46
47                 if (move_uploaded_file($_FILES['logo']['tmp_name'], $uploadPath)) {
48                     $logoPath = 'uploads/institution/' . $filename;
49
50                     // Eliminar logo anterior si existe
51                     $current = $this->institutionModel->getId(1);
52                     if ($current['logo_path'] && file_exists(BASE_PATH . '/' . $current['logo_path'])) {
53                         unlink(BASE_PATH . '/' . $current['logo_path']);
54                     }
55                 }
56             }
57
58             $data = [
59                 'name' => Security::sanitize($_POST['name']),
60                 'address' => Security::sanitize($_POST['address']),
61                 'province' => Security::sanitize($_POST['province']),
62                 'city' => Security::sanitize($_POST['city']),
63                 'phone' => Security::sanitize($_POST['phone']),
64                 'email' => Security::sanitize($_POST['email']),
65                 'director_name' => Security::sanitize($_POST['director_name']),
66                 'amie_code' => Security::sanitize($_POST['amie_code']),
67                 'website' => Security::sanitize($_POST['website']),
68                 'logo_path' => $logoPath
69             ];
70
71             $this->institutionModel->update(1, $data);
72             header('Location: ?action=institution&success=1');
73             exit;
74         }
75     }
```

```

76
77 public function assignShift() {
78     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
79         $data = [
80             ':institution_id' => 1,
81             ':shift_id' => (int)$_POST['shift_id']
82         ];
83
84         $this->institutionShiftModel->assign($data);
85         header('Location: ?action=institution&shift_assigned=1');
86         exit;
87     }
88 }
89
90 public function removeShift() {
91     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
92         $shiftId = (int)$_POST['shift_id'];
93         $this->institutionShiftModel->remove(1, $shiftId);
94         header('Location: ?action=institution&shift_removed=1');
95         exit;
96     }
97 }
98 }

```

## 13. JustificationController.php

Ruta: 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/Notification.php';
9
10 class JustificationController {
11     private $justificationModel;
12     private $attendanceModel;
13     private $notificationModel;
14
15     public function __construct() {
16         Security::requireLogin();
17
18         $db = new Database();
19         $this->justificationModel = new Justification($db);
20         $this->attendanceModel = new Attendance($db);
21         $this->notificationModel = new Notification($db);
22     }
23
24     public function submit() {
25         if (!Security::hasRole(['estudiante', 'representante'])) {
26             die('Acceso denegado');
27         }
28
29         $attendanceId = (int)$$_GET['attendance_id'];
30
31         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
32             if ($this->justificationModel->existsByAttendance($attendanceId)) {
33                 header('Location: ?action=my_attendance&error=already_justified');
34                 exit;
35             }
36
37             $documentPath = null;
38
39             if (isset($_FILES['document']) && $_FILES['document']['error'] == 0) {
40                 $uploadDir = BASE_PATH . '/uploads/justifications/';
41                 if (!file_exists($uploadDir)) {
42                     mkdir($uploadDir, 0755, true);
43                 }
44                 $extension = pathinfo($_FILES['document']['name'], PATHINFO_EXTENSION);
45                 $filename = uniqid() . '.' . $extension;
46                 if (move_uploaded_file($_FILES['document']['tmp_name'], $uploadDir . $filename)) {
47                     $documentPath = 'uploads/justifications/' . $filename;
48                 }
49             }
50
51             $studentId = Security::hasRole('estudiante')
52                 ? $_SESSION['user_id']
53                 : (int)$$_POST['student_id'];
54
55             $data = [
56                 'attendance_id' => $attendanceId,
57                 'student_id' => $studentId,
58                 'submitted_by' => $_SESSION['user_id'],
59                 'reason' => Security::sanitize($_POST['reason']),
60                 'document_path' => $documentPath
61             ];
62
63             $this->justificationModel->create($data);
64
65             // Notificar a autoridades/inspectores - enlace directo a pendientes
66             $this->_notifyReviewers(
67                 '■ Nueva justificación pendiente',
68                 'Un estudiante envió una justificación que requiere revisión.',
69                 'justificacion',
70                 '?action=pending_justifications'
71             );
72
73             header('Location: ?action=my_justifications&success=1');
74             exit;
75         }
76     }
77 }
```



```

76
77     if ($this->justificationModel->existsByAttendance($attendanceId)) {
78         header('Location: ?action=my_attendance&error=already_justified');
79         exit;
80     }
81
82     include BASE_PATH . '/views/justifications/submit.php';
83 }
84
85 public function myJustifications() {
86     if (!Security::hasRole(['estudiante', 'representante'])) {
87         die('Acceso denegado');
88     }
89     $studentId = $_SESSION['user_id'];
90     $justifications = $this->justificationModel->getByStudent($studentId);
91     include BASE_PATH . '/views/justifications/my_list.php';
92 }
93
94 public function pending() {
95     if (!Security::hasRole(['autoridad', 'inspector'])) {
96         die('Acceso denegado');
97     }
98     $justifications = $this->justificationModel->getPending();
99     include BASE_PATH . '/views/justifications/pending.php';
100 }
101
102 public function review() {
103     if (!Security::hasRole(['autoridad', 'inspector'])) {
104         die('Acceso denegado');
105     }
106
107     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
108         $justificationId = (int)$_POST['justification_id'];
109         $action = $_POST['review_action'];
110         $notes = Security::sanitize($_POST['notes'] ?? '');
111
112         // Obtener datos ANTES de actualizar para poder notificar
113         $justification = $this->justificationModel->getById($justificationId);
114
115         if ($action === 'approve') {
116             $this->justificationModel->approve($justificationId, $_SESSION['user_id'], $notes);
117
118             if ($justification) {
119                 // Notificar al estudiante - enlace a sus justificaciones
120                 $this->notificationModel->create(
121                     $justification['student_id'],
122                     '■ Justificación aprobada',
123                     'Tu justificación fue aprobada.' . ($notes ? " Nota: $notes" : ''),
124                     'success',
125                     '?action=my_justifications'
126                 );
127                 // Notificar al representante si fue distinto del estudiante
128                 if ($justification['submitted_by'] != $justification['student_id']) {
129                     $this->notificationModel->create(
130                         $justification['submitted_by'],
131                         '■ Justificación aprobada',
132                         'La justificación que enviaste fue aprobada.' . ($notes ? " Nota: $notes" : ''),
133                         'success',
134                         '?action=my_justifications'
135                     );
136                 }
137             }
138         } else {
139             $this->justificationModel->reject($justificationId, $_SESSION['user_id'], $notes);
140
141             if ($justification) {
142                 // Notificar al estudiante - puede enviar una nueva justificación
143                 $this->notificationModel->create(
144                     $justification['student_id'],
145                     '■ Justificación rechazada',
146                     'Tu justificación fue rechazada.' . ($notes ? " Motivo: $notes" : ''),
147                     'danger',
148                     '?action=my_justifications'
149                 );
150                 if ($justification['submitted_by'] != $justification['student_id']) {
151                     $this->notificationModel->create(
152                         $justification['submitted_by'],
153                         '■ Justificación rechazada',
154                         'La justificación que enviaste fue rechazada.' . ($notes ? " Motivo: $notes" : ''),

```

```

155         'danger',
156         '?action=my_justifications'
157     );
158     }
159 }
160 }

161
162     header('Location: ?action=pending_justifications&success=1');
163     exit;
164 }
165 }
166
167 public function reviewed() {
168     if (!Security::hasRole(['autoridad', 'inspector'])) {
169         die('Acceso denegado');
170     }
171     $filter = $_GET['filter'] ?? 'all';
172     $justifications = $this->justificationModel->getReviewed();
173
174     if ($filter !== 'all') {
175         $justifications = array_values(array_filter($justifications, function($j) use ($filter) {
176             return $j['status'] === $filter;
177         }));
178     }
179
180     include BASE_PATH . '/views/justifications/reviewed.php';
181 }
182
183 // Privado: notificar a todos los revisores
184 private function _notifyReviewers($title, $message, $type = 'info', $link = null) {
185     $db = new Database();
186     $pdo = $db->connect();
187
188     $sql = "SELECT DISTINCT u.id FROM users u
189         INNER JOIN user_roles ur ON u.id = ur.user_id
190         INNER JOIN roles r ON ur.role_id = r.id
191         WHERE r.name IN ('autoridad','inspector') AND u.id != :sender";
192     $stmt = $pdo->prepare($sql);
193     $stmt->execute([':sender' => $_SESSION['user_id']]);
194     $reviewers = $stmt->fetchAll(PDO::FETCH_COLUMN);
195
196     foreach ($reviewers as $reviewerId) {
197         $this->notificationModel->create($reviewerId, $title, $message, $type, $link);
198     }
199 }
200 }

```

## 14. NotificationController.php

Ruta: 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 }
```

## 15. ProfileController.php

Ruta: 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  }
```

## 16. ReportController.php

Ruta: 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                     <td>' . $row['student_name'] . '</td>
119                     <td>' . $row['subject_name'] . '</td>
120                     <td>' . $row['hour_period'] . '</td>
121                     <td style="background-color: ' . $statusColor . ';">' . ucfirst($row['status']) . '</td>
122                     <td>' . ($row['observation'] ?? '-') . '</td>
123                 </tr>';
124             }
125
126     $html .= '</tbody></table>';
127
128     $pdf->setFont('helvetica', '', 9);
129
130     $pdf->writeHTML($html, true, false, true, false, '');
131
132     // Generar nombre de archivo con curso
133     $courseName = $this->sanitizeFilename($course['name']);
134     $filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.pdf';
135
136     $pdf->Output($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 = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
146     $course = $this->getCourseInfo($courseId);
147     $institution = $this->institutionModel->getById($_SESSION['institution_id']);
148
149     $spreadsheet = new Spreadsheet();
150     $sheet = $spreadsheet->getActiveSheet();
151
152     $sheet->setTitle('Asistencia');
153
154     // Encabezado

```

```

155 $sheet->mergeCells('A1:F1');
156 $sheet->setCellValue('A1', 'REPORTE DE ASISTENCIA');
157 $sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
158 $sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);
159
160 $sheet->mergeCells('A2:F2');
161
162 $sheet->setCellValue('A2', $institution['name'] ?? 'Institución Educativa');
163 $sheet->getStyle('A2')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);
164
165 $sheet->setCellValue('A4', 'Curso:');
166 $sheet->setCellValue('B4', html_entity_decode($course['name'], ENT_QUOTES, 'UTF-8'));
167 $sheet->getStyle('A4')->getFont()->setBold(true);
168
169 $sheet->setCellValue('A5', 'Período:');
170 $sheet->setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate))...);
171 $sheet->getStyle('A5')->getFont()->setBold(true);
172
173 // Cabecera de tabla
174 $row = 7;
175 $headers = ['Fecha', 'Estudiante', 'Asignatura', 'Hora', 'Estado', 'Observación'];
176 $col = 'A';
177
178 foreach ($headers as $header) {
179     $sheet->setCellValue($col . $row, $header);
180     $sheet->getStyle($col . $row)->getFont()->setBold(true);
181     $sheet->getStyle($col . $row)->getFill()
182         ->setFillType(Fill::FILL_SOLID)
183         ->getStartColor()->setRGB('007bff');
184     $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
185     $col++;
186 }
187
188 // Datos
189 $row = 8;
190 foreach ($data as $record) {
191     $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
192     $sheet->setCellValue('B' . $row, $record['student_name']);
193     $sheet->setCellValue('C' . $row, $record['subject_name']);
194     $sheet->setCellValue('D' . $row, $record['hour_period']);
195     $sheet->setCellValue('E' . $row, ucfirst($record['status']));
196     $sheet->setCellValue('F' . $row, $record['observation'] ? '-' : '');
197
198     $statusColor = $this->getStatusColorHex($record['status']);
199     $sheet->getStyle('E' . $row)->getFill()
200         ->setFillType(Fill::FILL_SOLID)
201         ->getStartColor()->setRGB($statusColor);
202
203     $row++;
204 }
205
206 // Ajustar anchos
207 $sheet->getColumnDimension('A')->setWidth(12);
208 $sheet->getColumnDimension('B')->setWidth(25);
209 $sheet->getColumnDimension('C')->setWidth(20);
210 $sheet->getColumnDimension('D')->setWidth(15);
211 $sheet->getColumnDimension('E')->setWidth(12);
212 $sheet->getColumnDimension('F')->setWidth(25);
213
214 // Bordes
215 $styleArray = [
216     'borders' => [
217         'allBorders' => [
218             'borderStyle' => Border::BORDER_THIN,
219         ],
220     ],
221 ];
222 $sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);
223
224 $writer = new \PhpOffice\PhpSpreadsheet\Writer\Xlsx($spreadsheet);
225
226 // Generar nombre de archivo con curso
227 $courseName = $this->sanitizeFilename($course['name']);
228 $filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.xlsx';
229
230 header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
231 header('Content-Disposition: attachment;filename="' . $filename . '"');
232 header('Cache-Control: max-age=0');
233
234 $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
242             return $course;
243         }
244     }
245     return null;
246 }
247
248 private function getStatusColor($status) {
249     switch ($status) {
250         case 'presente': return '#d4edda';
251         case 'ausente': return '#f8d7da';
252         case 'tardanza': return '#fff3cd';
253         case 'justificado': return '#d1ecf1';
254         default: return '#ffffff';
255     }
256 }
257
258 private function getStatusColorHex($status) {
259     switch ($status) {
260         case 'presente': return '28a745';
261         case 'ausente': return 'dc3545';
262         case 'tardanza': return 'ffc107';
263         case 'justificado': return '17a2b8';
264         default: return 'FFFFFF';
265     }
266 }
267
268 private function sanitizeFilename($filename) {
269     // Decodificar HTML entities
270     $filename = html_entity_decode($filename, ENT_QUOTES, 'UTF-8');
271
272     // Reemplazar caracteres no permitidos en nombres de archivo
273     $filename = str_replace(['"', '/', '\\', ':', '*', '?', '<', '>', '|'], '', $filename);
274
275     // Reemplazar espacios con guiones bajos
276     $filename = str_replace(' ', '_', $filename);
277
278     // Reemplazar guiones múltiples
279     $filename = preg_replace('/-+/', '-', $filename);
280
281     // Limitar longitud (máximo 50 caracteres)
282     if (strlen($filename) > 50) {
283         $filename = substr($filename, 0, 50);
284     }
285
286     // Limpiar caracteres al inicio y final
287     $filename = trim($filename, '-_');
288
289     return $filename;
290 }

```

## 17. ReportController2.php

Ruta: 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
82 $pdf->Cell(0, 6, date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)), 0, 1...
83 $pdf->Ln(3);
84
85 $html = '<table border="1" cellpadding="4">';
86     <thead>
87         <tr style="background-color: #007bff; color: white; font-weight: bold;">
88             <th width="10%">Fecha</th>
89             <th width="25%">Estudiante</th>
90             <th width="20%">Asignatura</th>
91             <th width="15%">Hora</th>
92             <th width="12%">Estado</th>
93             <th width="18%">Observación</th>
94         </tr>
95     </thead>
96     <tbody>
97         foreach ($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')->fontWeight->setBold(true)->setSize(16);
139         $sheet->getStyle('A1')->horizontalAlignment->setHorizontal(Alignment::HORIZONTAL_CENTER);
140
141         $sheet->mergeCells('A2:F2');
142         $sheet->setCellValue('A2', 'Unidad Educativa Demo');
143         $sheet->getStyle('A2')->horizontalAlignment->setHorizontal(Alignment::HORIZONTAL_CENTER);
144
145         $sheet->setCellValue('A4', 'Curso:');
146         $sheet->setCellValue('B4', $course['name'] . ' - ' . $course['shift_name']);
147         $sheet->getStyle('A4')->fontWeight->setBold(true);
148
149         $sheet->setCellValue('A5', 'Período:');
150         $sheet->setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($enda...
151         $sheet->getStyle('A5')->fontWeight->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
162         $sheet->getStyle($col . $row)->getFill()
163             ->setFillType(Fill::FILL_SOLID)
164             ->getStartColor()->setRGB('007bff');
165         $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
166         $col++;
167     }
168
169     // Datos
170     $row = 8;
171     foreach ($data as $record) {
172         $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
173         $sheet->setCellValue('B' . $row, $record['student_name']);
174         $sheet->setCellValue('C' . $row, $record['subject_name']);
175         $sheet->setCellValue('D' . $row, $record['hour_period']);
176         $sheet->setCellValue('E' . $row, ucfirst($record['status']));
177         $sheet->setCellValue('F' . $row, $record['observation'] ? '-' : '');
178
179         $statusColor = $this->getStatusColorHex($record['status']);
180         $sheet->getStyle('E' . $row)->getFill()
181             ->setFillType(Fill::FILL_SOLID)
182             ->getStartColor()->setRGB($statusColor);
183
184         $row++;
185     }
186
187     // Ajustar anchos
188     $sheet->getColumnDimension('A')->setWidth(12);
189     $sheet->getColumnDimension('B')->setWidth(25);
190     $sheet->getColumnDimension('C')->setWidth(20);
191     $sheet->getColumnDimension('D')->setWidth(15);
192     $sheet->getColumnDimension('E')->setWidth(12);
193     $sheet->getColumnDimension('F')->setWidth(25);
194
195     // Bordes
196     $styleArray = [
197         'borders' => [
198             'allBorders' => [
199                 'borderStyle' => Border::BORDER_THIN,
200             ],
201         ],
202     ];
203     $sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);
204
205     $writer = new Xlsx($spreadsheet);
206
207     header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
208     header('Content-Disposition: attachment; filename="reporte_asistencia_' . date('YmdHis') . '.xlsx"');
209     header('Cache-Control: max-age=0');
210
211     $writer->save('php://output');
212     exit;
213
214 } catch (Exception $e) {
215     die('Error generando Excel: ' . $e->getMessage());
216 }
217
218 private function getCourseInfo($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
242         case 'ausente': return 'f8d7da';
243         case 'tardanza': return 'fff3cd';
244         case 'justificado': return 'd1ecf1';
245         default: return 'FFFFFF';
246     }
247 }
```

## 18. RepresentativeController.php

Ruta: 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 }
```

```

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 }

```

## 19. ScheduleController.php

Ruta: 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['error']));
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 }
```



```

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 }

```

## 20. SearchController.php

Ruta: 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 }
```



## 21. StatsController.php

Ruta: 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 }

```

## 22. TutorController.php

Ruta: 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         $stopAbsences = $this->attendanceModel->getTutorTopAbsences($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) {
59             echo json_encode(['error' => 'No es tutor']);
60             exit;
61         }
62
63         $filters = [];
64         if (!empty($_GET['subject_id'])) $filters['subject_id'] = (int)$_GET['subject_id'];
65         if (!empty($_GET['student_id'])) $filters['student_id'] = (int)$_GET['student_id'];
66         if (!empty($_GET['status'])) $filters['status'] = $_GET['status'];
67         if (!empty($_GET['start_date'])) $filters['start_date'] = $_GET['start_date'];
68         if (!empty($_GET['end_date'])) $filters['end_date'] = $_GET['end_date'];
69
70         $attendances = $this->attendanceModel->getTutorCourseAttendance($courseId, $filters);
71         $stats = $this->attendanceModel->getTutorCourseStats($courseId, $filters);
72
73         echo json_encode([
74             'stats' => $stats,
75             'attendances' => $attendances,
```

```
76         'total'      => count($attendances),
77     });
78     exit;
79 }
80 }
```



## 23. UserController.php

Ruta: 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         $errors[] = "El email ya está registrado";
101     }
102
103     if (!empty($errors)) {
104         $roles = $this->roleModel->getAll();
105         include BASE_PATH . '/views/users/create.php';
106         return;
107     }
108
109     // Crear usuario
110     $userData = [
111         'institution_id' => $_SESSION['institution_id'],
112         'username' => Security::sanitize($_POST['username']),
113         'email' => Security::sanitize($_POST['email']),
114         'password' => $_POST['password'],
115         'first_name' => Security::sanitize($_POST['first_name']),
116         'last_name' => Security::sanitize($_POST['last_name']),
117         'dni' => !empty($_POST['dni']) ? Security::sanitize($_POST['dni']) : null,
118         'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null
119     ];
120
121     if ($this->userModel->create($userData)) {
122         // Obtener ID del usuario recién creado
123         $newUserId = $this->userModel->db->lastInsertId();
124
125         // Asignar roles si se seleccionaron
126         if (!empty($_POST['roles'])) {
127             foreach ($_POST['roles'] as $roleId) {
128                 $this->userModel->assignRole($newUserId, (int)$roleId);
129             }
130         }
131
132         header('Location: ?action=users&created=1');
133         exit;
134     } else {
135         $errors[] = "Error al crear el usuario";
136         $roles = $this->roleModel->getAll();
137         include BASE_PATH . '/views/users/create.php';
138     }
139 }
140
141 public function edit() {
142     $userId = (int)$_GET['id'];
143     $user = $this->userModel->findById($userId);
144
145     if (!$user) {
146         header('Location: ?action=users&error=not_found');
147         exit;
148     }
149
150     // Verificar que pertenece a la misma institución
151     if ($user['institution_id'] !== $_SESSION['institution_id']) {
152         die('Acceso denegado');
153     }
154 }

```

```

155
156 if ($_SERVER['REQUEST_METHOD'] === 'GET') {
157     $roles = $this->roleModel->getAll();
158     $userRoles = $this->userModel->getUserRoleIds($userId);
159     include BASE_PATH . '/views/users/edit.php';
160     return;
161 }
162
163 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
164     $errors = [];
165
166     // Validaciones
167     if (empty($_POST['username'])) {
168         $errors[] = "El nombre de usuario es obligatorio";
169     }
170     if (empty($_POST['email']) || !filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
171         $errors[] = "Email inválido";
172     }
173     if (empty($_POST['first_name'])) {
174         $errors[] = "El nombre es obligatorio";
175     }
176     if (empty($_POST['last_name'])) {
177         $errors[] = "El apellido es obligatorio";
178     }
179
180     // Verificar username único (excepto el actual)
181     $existingUser = $this->userModel->findByUsername($_POST['username']);
182     if ($existingUser && $existingUser['id'] != $userId) {
183         $errors[] = "El nombre de usuario ya está registrado";
184     }
185
186     // Verificar email único (excepto el actual)
187     $existingEmail = $this->userModel->findByEmail($_POST['email']);
188     if ($existingEmail && $existingEmail['id'] != $userId) {
189         $errors[] = "El email ya está registrado";
190     }
191
192     // Si hay nueva contraseña, validarla
193     if (!empty($_POST['password'])) {
194         if (strlen($_POST['password']) < 6) {
195             $errors[] = "La contraseña debe tener al menos 6 caracteres";
196         }
197         if ($_POST['password'] != $_POST['confirm_password']) {
198             $errors[] = "Las contraseñas no coinciden";
199         }
200     }
201
202     if (!empty($errors)) {
203         $roles = $this->roleModel->getAll();
204         $userRoles = $this->userModel->getUserRoleIds($userId);
205         include BASE_PATH . '/views/users/edit.php';
206         return;
207     }
208
209     // Actualizar usuario
210     $updateData = [
211         'id' => $userId,
212         'username' => Security::sanitize($_POST['username']),
213         'email' => Security::sanitize($_POST['email']),
214         'first_name' => Security::sanitize($_POST['first_name']),
215         'last_name' => Security::sanitize($_POST['last_name']),
216         'dni' => !empty($_POST['dni']) ? Security::sanitize($_POST['dni']) : null,
217         'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null
218     ];
219
220     // Solo actualizar contraseña si se proporcionó una nueva
221     if (!empty($_POST['password'])) {
222         $updateData['password'] = $_POST['password'];
223     }
224
225     if ($this->userModel->update($updateData)) {
226         header('Location: ?action=users&updated=1');
227         exit;
228     } else {
229         $errors[] = "Error al actualizar el usuario";
230         $roles = $this->roleModel->getAll();
231         $userRoles = $this->userModel->getUserRoleIds($userId);
232         include BASE_PATH . '/views/users/edit.php';
233     }

```

```

234     }
235 }
236
237 public function delete() {
238     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
239         $userId = (int)$_POST['user_id'];
240         $user = $this->userModel->findById($userId);
241
242         if (!$user) {
243             header('Location: ?action=users&error=not_found');
244             exit;
245         }
246
247         // Verificar que pertenece a la misma institución
248         if ($user['institution_id'] != $_SESSION['institution_id']) {
249             die('Acceso denegado');
250         }
251
252         // No permitir eliminar al propio usuario
253         if ($userId == $_SESSION['user_id']) {
254             header('Location: ?action=users&error=self_delete');
255             exit;
256         }
257
258         // Verificar si tiene asistencias registradas
259         $db = new Database();
260         $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM attendances WHERE student_id = :user_id...");
261         $stmt->execute([':user_id' => $userId]);
262         $result = $stmt->fetch();
263
264         if ($result['count'] > 0) {
265             // Soft delete - desactivar en lugar de eliminar
266             if ($this->userModel->deactivate($userId)) {
267                 header('Location: ?action=users&deactivated=1');
268                 exit;
269             }
270         } else {
271             // Hard delete - eliminar completamente
272             if ($this->userModel->delete($userId)) {
273                 header('Location: ?action=users&deleted=1');
274                 exit;
275             }
276         }
277
278         header('Location: ?action=users&error=delete_failed');
279         exit;
280     }
281 }
282 }

```

## 24. Backup.php

Ruta: 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 > %s 2>&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 > %s 2>&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) > 0) {
52             return $filename;
53         }
54
55         // Si falló, intentar eliminar archivo vacío
56         if (file_exists($filepath)) {
57             unlink($filepath);
58         }
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 }
```

```

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 &lt;&lt; (10 * $pow));
91     return round($bytes, $precision) . ' ' . $units[$pow];
92 }
93
94 public function deleteOldBackups($daysToKeep = 30) {
95     $files = glob($this-&gt;backupDir . 'backup_*.sql');
96     $deleted = 0;
97
98     foreach ($files as $file) {
99         if (time() - filemtime($file) &gt; ($daysToKeep * 24 * 60 * 60)) {
100             unlink($file);
101             $deleted++;
102         }
103     }
104
105     return $deleted;
106 }
107
108 public function downloadBackup($filename) {
109     $filepath = $this-&gt;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-&gt;backupDir . $filename;
129
130     if (file_exists($filepath)) {
131         return unlink($filepath);
132     }
133
134     return false;
135 }
136 }

```

## 25. Logger.php

Ruta: 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 }
```

## 26. Mailer.php

Ruta: 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  }
```



## 27. Security.php

Ruta: 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($roleNames) {
38         if (!isset($_SESSION['roles'])) return false;
39         $roleNames = (array) $roleNames;
40         return count(array_intersect($roleNames, $_SESSION['roles'])) > 0;
41     }
42
43     public static function startSession($rememberMe = false) {
44         if ($rememberMe) {
45             // Sesión de 7 días
46             $lifetime = 7 * 24 * 60 * 60; // 7 días
47         } else {
48             // Sesión expira al cerrar navegador
49             $lifetime = 0;
50         }
51
52         session_set_cookie_params([
53             'lifetime' => $lifetime,
54             'path' => '/',
55             'httponly' => true,
56             'samesite' => 'Lax'
57         ]);
58
59         session_start();
60
61         // Regenerar ID por seguridad
62         if (!isset($_SESSION['initiated'])) {
63             session_regenerate_id(true);
64             $_SESSION['initiated'] = true;
65         }
66
67         // Timeout de inactividad (30 minutos)
68         if (isset($_SESSION['last_activity']) &&
69             (time() - $_SESSION['last_activity'] > 1800)) {
70             session_unset();
71             session_destroy();
72             return false;
73         }
74
75         $_SESSION['last_activity'] = time();
```

```
76         return true;
77     }
78 }
```

## 28. Attendance.php

Ruta: 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
82         FROM attendances a
83         INNER JOIN subjects s ON a.subject_id = s.id
84         INNER JOIN courses c ON a.course_id = c.id
85         INNER JOIN shifts sh ON a.shift_id = sh.id
86         WHERE a.student_id = :student_id
87         ORDER BY a.date DESC, a.hour_period
88         LIMIT 100";
89     $stmt = $this->db->prepare($sql);
90     $stmt->execute([':student_id' => $studentId]);
91     return $stmt->fetchAll();
92 }
93
94 public function getReportData($courseId, $startDate, $endDate) {
95     $sql = "SELECT a.*,
96         CONCAT(u.last_name, ' ', u.first_name) as student_name,
97         s.name as subject_name
98     FROM attendances a
99     INNER JOIN users u ON a.student_id = u.id
100    INNER JOIN subjects s ON a.subject_id = s.id
101    WHERE a.course_id = :course_id
102    AND a.date BETWEEN :start_date AND :end_date
103    ORDER BY a.date, u.last_name, u.first_name";
104    $stmt = $this->db->prepare($sql);
105    $stmt->execute([':course_id' => $courseId, ':start_date' => $startDate, ':end_date' => $endDate]);
106    return $stmt->fetchAll();
107 }
108
109 public function getStatsByCourse($courseId, $startDate, $endDate) {
110     $sql = "SELECT
111         COUNT(*) as total,
112         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
113         SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
114         SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,
115         SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
116     FROM attendances
117     WHERE course_id = :course_id AND date BETWEEN :start_date AND :end_date";
118     $stmt = $this->db->prepare($sql);
119     $stmt->execute([':course_id' => $courseId, ':start_date' => $startDate, ':end_date' => $endDate]);
120     return $stmt->fetch();
121 }
122
123 public function getFiltered($filters) {
124     $sql = "SELECT a.*,
125         CONCAT(u.last_name, ' ', u.first_name) as student_name,
126         c.name as course_name, s.name as subject_name,
127         CONCAT(t.last_name, ' ', t.first_name) as teacher_name
128     FROM attendances a
129     INNER JOIN users u ON a.student_id = u.id
130     INNER JOIN courses c ON a.course_id = c.id
131     INNER JOIN subjects s ON a.subject_id = s.id
132     INNER JOIN users t ON a.teacher_id = t.id
133     WHERE 1=1";
134     $params = [];
135     if (!empty($filters['course_id'])) { $sql .= " AND a.course_id = :course_id"; $params[':course_id'] = $f...
136     if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params[':student_id'] = $f...
137     if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params[':subject_id'] = $f...
138     if (!empty($filters['status'])) { $sql .= " AND a.status = :status"; $params[':status'] = $...
139     if (!empty($filters['start_date'])) { $sql .= " AND a.date >= :start_date"; $params[':start_date'] = $...
140     if (!empty($filters['end_date'])) { $sql .= " AND a.date <= :end_date"; $params[':end_date'] = $...
141     $sql .= " ORDER BY a.date DESC, u.last_name, u.first_name";
142     $stmt = $this->db->prepare($sql);
143     $stmt->execute($params);
144     return $stmt->fetchAll();
145 }
146
147 public function getDayStats($courseId, $date) {
148     $sql = "SELECT COUNT(*) as total,
149         SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
150         SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
151         SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza
152     FROM attendances WHERE course_id = :course_id AND date = :date";
153     $stmt = $this->db->prepare($sql);
154     $stmt->execute([':course_id' => $courseId, ':date' => $date]);
155     return $stmt->fetch();

```

[illegible]

```

234     $params = ['course_id' => $courseId];
235
236     if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params['subject_id'] = $fille...
237     if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params['student_id'] = $fille...
238     if (!empty($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 }

```

## 29. ClassSchedule.php

Ruta: 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 }

```



## 30. Course.php

Ruta: 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
82         $sql = "SELECT c.*, s.name as shift_name
83             FROM courses c
84             INNER JOIN shifts s ON c.shift_id = s.id
85             WHERE c.id = :id";
86
87         $stmt = $this->db->prepare($sql);
88         $stmt->execute([':id' => $id]);
89         return $stmt->fetch();
90     }
91
92     public function update($data) {
93         $sql = "UPDATE courses SET
94             name = :name,
95             grade_level = :grade_level,
96             parallel = :parallel,
97             shift_id = :shift_id,
98             updated_at = NOW()
99             WHERE id = :id";
100
101         $stmt = $this->db->prepare($sql);
102         return $stmt->execute($data);
103     }
104
105     public function delete($id) {
106         // Eliminar relaciones primero
107         $sql = "DELETE FROM course_students WHERE course_id = :id";
108         $stmt = $this->db->prepare($sql);
109         $stmt->execute([':id' => $id]);
110
111         // Eliminar horarios
112         $sql = "DELETE FROM class_schedule WHERE course_id = :id";
113         $stmt = $this->db->prepare($sql);
114         $stmt->execute([':id' => $id]);
115
116         // Eliminar curso
117         $sql = "DELETE FROM courses WHERE id = :id";
118         $stmt = $this->db->prepare($sql);
119         $stmt->execute([':id' => $id]);
120     }
121
122     public function unenrollStudent($studentId, $schoolYearId) {
123         $sql = "DELETE FROM course_students
124             WHERE student_id = :student_id AND school_year_id = :school_year_id";
125
126         $stmt = $this->db->prepare($sql);
127         return $stmt->execute([
128             ':student_id' => $studentId,
129             ':school_year_id' => $schoolYearId
130         ]);
131     }

```

## 31. Institution.php

Ruta: 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 }
```

## 32. InstitutionShift.php

Ruta: 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 }
```

## 33. Justification.php

Ruta: 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 $result['count'] &gt; 0;
108 }
109
110 public function getId($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]);
118     return $stmt->fetch();
119 }
120
121 public function getReviewed($institutionId = null) {
122     $sql = "SELECT j.*,
123         CONCAT(u.last_name, ' ', u.first_name) as student_name,
124         CONCAT(s.last_name, ' ', s.first_name) as submitted_by_name,
125         CONCAT(r.last_name, ' ', r.first_name) as reviewer_name,
126         a.date as attendance_date,
127         a.hour_period,
128         sub.name as subject_name,
129         c.name as course_name
130         FROM justifications j
131         INNER JOIN users u ON j.student_id = u.id
132         INNER JOIN users s ON j.submitted_by = s.id
133         INNER JOIN attendances a ON j.attendance_id = a.id
134         INNER JOIN subjects sub ON a.subject_id = sub.id
135         LEFT JOIN users r ON j.reviewed_by = r.id
136         LEFT JOIN course_students cs ON u.id = cs.student_id
137         LEFT JOIN courses c ON cs.course_id = c.id
138         WHERE j.status IN ('aprobado', 'rechazado')
139         ORDER BY j.updated_at DESC";
140
141     $stmt = $this->db->query($sql);
142     return $stmt->fetchAll();
143 }
144 }

```

## 34. Notification.php

Ruta: 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
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 }

```



## 35. Representative.php

Ruta: 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 }
```

## 36. Role.php

Ruta: 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 }
```

## 37. SchoolYear.php

Ruta: 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 $stmt->fetchAll();
27     }
28
29     public function findById($id) {
30         $sql = "SELECT * FROM school_years
31             WHERE id = :id AND institution_id = :institution_id";
32
33         $stmt = $this->db->prepare($sql);
34         $stmt->execute([
35             ':id' => $id,
36             ':institution_id' => $_SESSION['institution_id']
37         ]);
38
39         return $stmt->fetch();
40     }
41
42     public function create($data) {
43         $sql = "INSERT INTO school_years
44             (institution_id, name, start_date, end_date, is_active)
45             VALUES
46             (:institution_id, :name, :start_date, :end_date, :is_active)";
47
48         $stmt = $this->db->prepare($sql);
49         return $stmt->execute([
50             ':institution_id' => $data['institution_id'],
51             ':name' => $data['name'],
52             ':start_date' => $data['start_date'],
53             ':end_date' => $data['end_date'],
54             ':is_active' => $data['is_active']
55         ]);
56     }
57
58     public function update($data) {
59         $sql = "UPDATE school_years
60             SET name=:name,
61                 start_date=:start_date,
62                 end_date=:end_date
63             WHERE id=:id
64                 AND institution_id=:institution_id";
65
66         $stmt = $this->db->prepare($sql);
67
68         return $stmt->execute([
69             ':id'=>$data['id'],
70             ':name'=>$data['name'],
71             ':start_date'=>$data['start_date'],
72             ':end_date'=>$data['end_date'],
73             ':institution_id'=>$_SESSION['institution_id']
74         ]);
75     }
```

```

76     }
77
78     public function delete($id) {
79         $stmt = $this->db->prepare(
80             "DELETE FROM school_years
81
82                 WHERE id=:id AND institution_id=:institution_id"
83             );
84
85         return $stmt->execute([
86             ':id'=>$id,
87             ':institution_id'=>$SESSION['institution_id']
88         ]);
89     }
90
91     public function activate($id) {
92         // desactiva todos
93         $this->db->prepare(
94             "UPDATE school_years
95                 SET is_active=0
96                 WHERE institution_id=:institution_id"
97         )->execute([':institution_id'=>$SESSION['institution_id']]);
98
99         if ($id == 0) return true;
100
101         // activa uno
102         $stmt = $this->db->prepare(
103             "UPDATE school_years
104                 SET is_active=1
105                 WHERE id=:id AND institution_id=:institution_id"
106         );
107
108         return $stmt->execute([
109             ':id'=>$id,
110             ':institution_id'=>$SESSION['institution_id']
111         ]);
112     }
113
114     public function deactivate($id) {
115         $stmt = $this->db->prepare(
116             "UPDATE school_years
117                 SET is_active=0
118                 WHERE id=:id AND institution_id=:institution_id"
119         );
120
121         return $stmt->execute([
122             ':id'=>$id,
123             ':institution_id'=>$SESSION['institution_id']
124         ]);
125     }
126
127     public function checkOverlap($start, $end, $excludeId = null) {
128
129         $sql = "SELECT COUNT(*)
130             FROM school_years
131             WHERE institution_id = :institution_id
132             AND (start_date <= :end AND end_date >= :start)";
133
134         if ($excludeId) {
135             $sql .= " AND id != :excludeId";
136         }
137
138         $stmt = $this->db->prepare($sql);
139
140         $params = [
141             ':institution_id' => $SESSION['institution_id'],
142             ':start' => $start,
143             ':end' => $end
144         ];
145
146         if ($excludeId) {
147             $params[':excludeId'] = $excludeId;
148         }
149
150         $stmt->execute($params);
151         return $stmt->fetchColumn() > 0;
152     }
153 }
154 }

```



## 38. Shift.php

Ruta: 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 }
```

## 39. Subject.php

Ruta: 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 }
```

## 40. TeacherAssignment.php

Ruta: 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
82             INNER JOIN users u ON ta.teacher_id = u.id
83             WHERE ta.course_id = :course_id
84             AND ta.school_year_id = :school_year_id
85             AND ta.is_tutor = 1
86             LIMIT 1";
87
88         $stmt = $this->db->prepare($sql);
89         $stmt->execute([
90             ':course_id' => $courseId,
91             ':school_year_id' => $schoolYearId
92         ]);
93         return $stmt->fetch();
94     }
95
96     public function setTutor($courseId, $teacherId, $schoolYearId) {
97         // Verificar si el docente ya es tutor de otro curso en el mismo año lectivo
98         $checkSql = "SELECT c.name
99             FROM teacher_assignments ta
100             INNER JOIN courses c ON ta.course_id = c.id
101             WHERE ta.teacher_id = :teacher_id
102             AND ta.school_year_id = :school_year_id
103             AND ta.is_tutor = 1
104             AND ta.course_id != :course_id";
105
106         $checkStmt = $this->db->prepare($checkSql);
107         $checkStmt->execute([
108             ':teacher_id' => $teacherId,
109             ':school_year_id' => $schoolYearId,
110             ':course_id' => $courseId
111         ]);
112
113         $existingTutor = $checkStmt->fetch();
114
115         if ($existingTutor) {
116             return [
117                 'success' => false,
118                 'message' => 'Este docente ya es tutor del curso: ' . $existingTutor['name']
119             ];
120         }
121
122         // Quitar tutor actual del curso
123         $sql1 = "UPDATE teacher_assignments
124             SET is_tutor = 0
125             WHERE course_id = :course_id AND school_year_id = :school_year_id";
126         $stmt1 = $this->db->prepare($sql1);
127         $stmt1->execute([
128             ':course_id' => $courseId,
129             ':school_year_id' => $schoolYearId
130         ]);
131
132         // Verificar si docente tiene asignación en el curso
133         $checkAssignment = "SELECT id FROM teacher_assignments
134             WHERE course_id = :course_id
135             AND teacher_id = :teacher_id
136             AND school_year_id = :school_year_id
137             LIMIT 1";
138
139         $stmt = $this->db->prepare($checkAssignment);
140         $stmt->execute([
141             ':course_id' => $courseId,
142             ':teacher_id' => $teacherId,
143             ':school_year_id' => $schoolYearId
144         ]);
145
146         $existing = $stmt->fetch();
147
148         if ($existing) {
149             // Actualizar asignación existente
150             $sql2 = "UPDATE teacher_assignments
151                 SET is_tutor = 1
152                 WHERE id = :id";
153             $stmt2 = $this->db->prepare($sql2);
154             $result = $stmt2->execute([':id' => $existing['id']]);
155         } else {
156             return [
157                 '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' => $result,
161
162     'message' => 'Tutor asignado correctamente'
163 ];
164 }
165
166 public function remove($assignmentId) {
167     // Verificar si es tutor antes de eliminar
168     $checkSql = "SELECT is_tutor FROM teacher_assignments WHERE id = :id";
169     $checkStmt = $this->db->prepare($checkSql);
170     $checkStmt->execute([':id' => $assignmentId]);
171     $assignment = $checkStmt->fetch();
172
173     if ($assignment && $assignment['is_tutor'] == 1) {
174         return [
175             'success' => false,
176             'message' => 'No se puede eliminar: este docente es tutor del curso. Quite primero la tutoría.'
177         ];
178     }
179
180     $sql = "DELETE FROM teacher_assignments WHERE id = :id";
181     $stmt = $this->db->prepare($sql);
182     $result = $stmt->execute([':id' => $assignmentId]);
183
184     return [
185         'success' => $result,
186         'message' => 'Asignación eliminada correctamente'
187     ];
188 }
189
190 public function getAll() {
191     $sql = "SELECT ta.*,
192         CONCAT(u.last_name, ' ', u.first_name) as teacher_name,
193         c.name as course_name,
194         s.name as subject_name,
195         sy.name as year_name
196     FROM teacher_assignments ta
197     INNER JOIN users u ON ta.teacher_id = u.id
198     INNER JOIN courses c ON ta.course_id = c.id
199     INNER JOIN subjects s ON ta.subject_id = s.id
200     INNER JOIN school_years sy ON ta.school_year_id = sy.id
201     ORDER BY c.name, s.name";
202
203     $stmt = $this->db->query($sql);
204     return $stmt->fetchAll();
205 }

```

## 41. User.php

Ruta: 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 }
```

```

76     });
77 }
78
79 public function getAll() {
80     $sql = "SELECT u.*, GROUP_CONCAT(r.name) as roles
81
82         FROM users u
83         LEFT JOIN user_roles ur ON u.id = ur.user_id
84         LEFT JOIN roles r ON ur.role_id = r.id
85         WHERE u.institution_id = :institution_id
86         GROUP BY u.id
87         ORDER BY u.last_name, u.first_name";
88
89     $stmt = $this->db->prepare($sql);
90     $stmt->execute([':institution_id' => $SESSION['institution_id']]);
91     return $stmt->fetchAll();
92 }
93
94 public function removeRole($userId, $roleId) {
95     $sql = "DELETE FROM user_roles WHERE user_id = :user_id AND role_id = :role_id";
96     $stmt = $this->db->prepare($sql);
97     return $stmt->execute([':user_id' => $userId, ':role_id' => $roleId]);
98 }
99
100 public function getUserWithRoles($userId) {
101     $user = $this->findById($userId);
102     if ($user) {
103         $user['roles'] = $this->getUserRoles($userId);
104     }
105     return $user;
106 }
107
108 /* public function getByRole($roleName) {
109     $sql = "SELECT u.* FROM users u
110         INNER JOIN user_roles ur ON u.id = ur.user_id
111         INNER JOIN roles r ON ur.role_id = r.id
112         WHERE r.name = :role_name AND u.institution_id = :institution_id
113         ORDER BY u.last_name, u.first_name";
114
115     $stmt = $this->db->prepare($sql);
116     $stmt->execute([
117         ':role_name' => $roleName,
118         ':institution_id' => $SESSION['institution_id']
119     ]);
120     return $stmt->fetchAll();
121 } */
122
123 // Agregar al final de la clase User
124
125 public function getByRole($roleName) {
126     $sql = "SELECT u.* FROM users u
127         INNER JOIN user_roles ur ON u.id = ur.user_id
128         INNER JOIN roles r ON ur.role_id = r.id
129         WHERE r.name = :role_name AND u.institution_id = :institution_id
130         ORDER BY u.last_name, u.first_name";
131
132     $stmt = $this->db->prepare($sql);
133     $stmt->execute([
134         ':role_name' => $roleName,
135         ':institution_id' => $SESSION['institution_id']
136     ]);
137     return $stmt->fetchAll();
138 }
139
140 public function getStudentsNotEnrolled($schoolYearId) {
141     $sql = "SELECT u.* FROM users u
142         INNER JOIN user_roles ur ON u.id = ur.user_id
143         INNER JOIN roles r ON ur.role_id = r.id
144         WHERE r.name = 'estudiante'
145         AND u.institution_id = :institution_id
146         AND u.id NOT IN (
147             SELECT student_id FROM course_students
148             WHERE school_year_id = :school_year_id
149         )
150         ORDER BY u.last_name, u.first_name";
151
152     $stmt = $this->db->prepare($sql);
153     $stmt->execute([
154         ':institution_id' => $SESSION['institution_id'],
155         ':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
162         FROM courses c
163         INNER JOIN course_students cs ON c.id = cs.course_id
164         INNER JOIN shifts s ON c.shift_id = s.id
165         WHERE cs.student_id = :student_id
166         AND cs.school_year_id = :school_year_id";
167
168     $stmt = $this->db->prepare($sql);
169     $stmt->execute([
170         ':student_id' => $studentId,
171         ':school_year_id' => $schoolYearId
172     ]);
173     return $stmt->fetch();
174 }
175
176 public function updateProfile($userId, $data) {
177     $sql = "UPDATE users
178         SET first_name = :first_name,
179         last_name = :last_name,
180         phone = :phone,
181         email = :email,
182         updated_at = NOW()
183         WHERE id = :id";
184
185     $stmt = $this->db->prepare($sql);
186     return $stmt->execute([
187         ':first_name' => $data['first_name'],
188         ':last_name' => $data['last_name'],
189         ':phone' => $data['phone'],
190         ':email' => $data['email'],
191         ':id' => $userId
192     ]);
193 }
194
195 public function updatePassword($userId, $newPassword) {
196     $sql = "UPDATE users SET password = :password WHERE id = :id";
197     $stmt = $this->db->prepare($sql);
198     return $stmt->execute([
199         ':password' => Security::hashPassword($newPassword),
200         ':id' => $userId
201     ]);
202 }
203
204 public function findByEmailOrUsername($emailOrUsername) {
205     $sql = "SELECT * FROM users
206         WHERE (email = :email OR username = :username)
207         AND is_active = 1";
208
209     $stmt = $this->db->prepare($sql);
210     $stmt->execute([
211         ':email' => $emailOrUsername,
212         ':username' => $emailOrUsername
213     ]);
214     return $stmt->fetch();
215 }
216
217 public function getUserRoleIds($userId) {
218     $sql = "SELECT r.id FROM roles r
219         INNER JOIN user_roles ur ON r.id = ur.role_id
220         WHERE ur.user_id = :user_id
221         ORDER BY r.name";
222
223     $stmt = $this->db->prepare($sql);
224     $stmt->execute([':user_id' => $userId]);
225     return $stmt->fetchAll(PDO::FETCH_COLUMN);
226 }
227
228 public function findByUsername($username) {
229     $sql = "SELECT * FROM users WHERE username = :username";
230     $stmt = $this->db->prepare($sql);
231     $stmt->execute([':username' => $username]);
232     return $stmt->fetch();
233 }
234
235 public function update($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
242         last_name = :last_name,
243         dni = :dni,
244         phone = :phone,
245         updated_at = NOW()
246         WHERE id = :id";
247
248     $params = [
249         ':id' => $data['id'],
250         ':username' => $data['username'],
251         ':email' => $data['email'],
252         ':password' => Security::hashPassword($data['password']),
253         ':first_name' => $data['first_name'],
254         ':last_name' => $data['last_name'],
255         ':dni' => $data['dni'],
256         ':phone' => $data['phone']
257     ];
258 } else {
259     // No actualizar password
260     $sql = "UPDATE users SET
261         username = :username,
262         email = :email,
263         first_name = :first_name,
264         last_name = :last_name,
265         dni = :dni,
266         phone = :phone,
267         updated_at = NOW()
268         WHERE id = :id";
269
270     $params = [
271         ':id' => $data['id'],
272         ':username' => $data['username'],
273         ':email' => $data['email'],
274         ':first_name' => $data['first_name'],
275         ':last_name' => $data['last_name'],
276         ':dni' => $data['dni'],
277         ':phone' => $data['phone']
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 }
```

## 42. index.php

Ruta: 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
82         require_once BASE_PATH . '/controllers/AcademicController.php';
83         $acadCtrl = new AcademicController();
84         $acadCtrl->index();
85         break;
86
87     case 'create_course':
88         require_once BASE_PATH . '/controllers/AcademicController.php';
89         $acadCtrl = new AcademicController();
90         $acadCtrl->createCourse();
91         break;
92
93     case 'create_subject':
94         require_once BASE_PATH . '/controllers/AcademicController.php';
95         $acadCtrl = new AcademicController();
96         $acadCtrl->createSubject();
97         break;
98
99     case 'enroll_students':
100         require_once BASE_PATH . '/controllers/AcademicController.php';
101         $acadCtrl = new AcademicController();
102         $acadCtrl->enrollStudents();
103         break;
104
105     case 'view_course_students':
106         require_once BASE_PATH . '/controllers/AcademicController.php';
107         $acadCtrl = new AcademicController();
108         $acadCtrl->viewCourseStudents();
109         break;
110
111     case 'reports':
112         require_once BASE_PATH . '/controllers/ReportController.php';
113         $reportCtrl = new ReportController();
114         $reportCtrl->index();
115         break;
116
117     case 'generate_pdf':
118         require_once BASE_PATH . '/controllers/ReportController.php';
119         $reportCtrl = new ReportController();
120         $reportCtrl->generatePDF();
121         break;
122
123     case 'generate_excel':
124         require_once BASE_PATH . '/controllers/ReportController.php';
125         $reportCtrl = new ReportController();
126         $reportCtrl->generateExcel();
127         break;
128
129     case 'manage_representatives':
130         require_once BASE_PATH . '/controllers/RepresentativeController.php';
131         $repCtrl = new RepresentativeController();
132         $repCtrl->manageRepresentatives();
133         break;
134
135     case 'my_children':
136         require_once BASE_PATH . '/controllers/RepresentativeController.php';
137         $repCtrl = new RepresentativeController();
138         $repCtrl->myChildren();
139         break;
140
141     case 'child_attendance':
142         require_once BASE_PATH . '/controllers/RepresentativeController.php';
143         $repCtrl = new RepresentativeController();
144         $repCtrl->childAttendance();
145         break;
146
147     case 'assignments':
148         require_once BASE_PATH . '/controllers/AssignmentController.php';
149         $assignCtrl = new AssignmentController();
150         $assignCtrl->index();
151         break;
152
153     case 'create_assignment':
154         require_once BASE_PATH . '/controllers/AssignmentController.php';
155         $assignCtrl = new AssignmentController();

```

```

155         $assignCtrl->assign();
156         break;
157
158     case 'set_tutor':
159         require_once BASE_PATH . '/controllers/AssignmentController.php';
160         $assignCtrl = new AssignmentController();
161
162         $assignCtrl->setTutor();
163         break;
164
165     case 'remove_tutor':
166         require_once BASE_PATH . '/controllers/AssignmentController.php';
167         $controller = new AssignmentController();
168         $controller->removeTutor();
169         break;
170
171     case 'get_course_teachers':
172         require_once BASE_PATH . '/controllers/AssignmentController.php';
173         $controller = new AssignmentController();
174         $controller->getCourseTeachers();
175         break;
176
177     case 'remove_assignment':
178         require_once BASE_PATH . '/controllers/AssignmentController.php';
179         $assignCtrl = new AssignmentController();
180         $assignCtrl->remove();
181         break;
182
183     case 'view_course_assignments':
184         require_once BASE_PATH . '/controllers/AssignmentController.php';
185         $assignCtrl = new AssignmentController();
186         $assignCtrl->viewByCourse();
187         break;
188
189     case 'profile':
190         require_once BASE_PATH . '/controllers/ProfileController.php';
191         $profileCtrl = new ProfileController();
192         $profileCtrl->view();
193         break;
194
195     case 'edit_profile':
196         require_once BASE_PATH . '/controllers/ProfileController.php';
197         $profileCtrl = new ProfileController();
198         $profileCtrl->edit();
199         break;
200
201     case 'change_password':
202         require_once BASE_PATH . '/controllers/ProfileController.php';
203         $profileCtrl = new ProfileController();
204         $profileCtrl->changePassword();
205         break;
206
207     case 'search_students':
208         require_once BASE_PATH . '/controllers/SearchController.php';
209         $searchCtrl = new SearchController();
210         $searchCtrl->searchStudents();
211         break;
212
213     case 'export_student_list':
214         require_once BASE_PATH . '/controllers/ExportController.php';
215         $exportCtrl = new ExportController();
216         $exportCtrl->exportStudentList();
217         break;
218
219     case 'stats':
220         require_once BASE_PATH . '/controllers/StatsController.php';
221         $statsCtrl = new StatsController();
222         $statsCtrl->index();
223         break;
224
225     case 'attendance_calendar':
226         require_once BASE_PATH . '/controllers/AttendanceController.php';
227         $attCtrl = new AttendanceController();
228         $attCtrl->calendar();
229         break;
230
231     case 'submit_justification':
232         require_once BASE_PATH . '/controllers/JustificationController.php';
233         $justCtrl = new JustificationController();
234         $justCtrl->submit();

```

```

234         break;
235
236     case 'my_justifications':
237         require_once BASE_PATH . '/controllers/JustificationController.php';
238         $justCtrl = new JustificationController();
239         $justCtrl->myJustifications();
240         break;
241
242     case 'pending_justifications':
243         require_once BASE_PATH . '/controllers/JustificationController.php';
244         $justCtrl = new JustificationController();
245         $justCtrl->pending();
246         break;
247
248     case 'reviewed_justifications':
249         require_once BASE_PATH . '/controllers/JustificationController.php';
250         $controller = new JustificationController();
251         $controller->reviewed();
252         break;
253
254     case 'review_justification':
255         require_once BASE_PATH . '/controllers/JustificationController.php';
256         $justCtrl = new JustificationController();
257         $justCtrl->review();
258         break;
259
260     case 'backups':
261         require_once BASE_PATH . '/controllers/BackupController.php';
262         $backupCtrl = new BackupController();
263         $backupCtrl->index();
264         break;
265
266     case 'create_backup':
267         require_once BASE_PATH . '/controllers/BackupController.php';
268         $backupCtrl = new BackupController();
269         $backupCtrl->create();
270         break;
271
272     case 'download_backup':
273         require_once BASE_PATH . '/controllers/BackupController.php';
274         $backupCtrl = new BackupController();
275         $backupCtrl->download();
276         break;
277
278     case 'cleanup_backups':
279         require_once BASE_PATH . '/controllers/BackupController.php';
280         $backupCtrl = new BackupController();
281         $backupCtrl->cleanup();
282         break;
283
284     case 'remove_role':
285         require_once BASE_PATH . '/controllers/UserController.php';
286         $userCtrl = new UserController();
287         $userCtrl->removeRole();
288         break;
289
290     case 'get_course_subjects':
291         require_once BASE_PATH . '/controllers/AttendanceController.php';
292         $controller = new AttendanceController();
293         $controller->getCourseSubjects();
294         break;
295
296     case 'get_teacher_course_subjects':
297         require_once BASE_PATH . '/controllers/AttendanceController.php';
298         $controller = new AttendanceController();
299         $controller->getTeacherCourseSubjects();
300         break;
301
302     case 'schedules':
303         require_once BASE_PATH . '/controllers/ScheduleController.php';
304         $controller = new ScheduleController();
305         $controller->index();
306         break;
307
308     case 'manage_schedule':
309         require_once BASE_PATH . '/controllers/ScheduleController.php';
310         $controller = new ScheduleController();
311         $controller->manageCourse();
312         break;

```

```
313
314 case 'delete_schedule_class':
315     require_once BASE_PATH . '/controllers/ScheduleController.php';
316     $controller = new ScheduleController();
317     $controller->deleteClass();
318     break;
319
320 case 'get_schedule_info':
321
322     require_once BASE_PATH . '/controllers/AttendanceController.php';
323     $controller = new AttendanceController();
324     $controller->getScheduleInfo();
325     break;
326
327 case 'get_existing_attendance':
328     require_once BASE_PATH . '/controllers/AttendanceController.php';
329     $controller = new AttendanceController();
330     $controller->getExistingAttendance();
331     break;
332
333 case 'tutor_management':
334     require_once BASE_PATH . '/controllers/AssignmentController.php';
335     $controller = new AssignmentController();
336     $controller->tutorManagement();
337     break;
338
339 case 'check_course_tutor':
340     require_once BASE_PATH . '/controllers/AssignmentController.php';
341     $controller = new AssignmentController();
342     $controller->checkCourseTutor();
343     break;
344
345 case 'get_course_subjects_schedule':
346     require_once BASE_PATH . '/controllers/ScheduleController.php';
347     $controller = new ScheduleController();
348     $controller->getCourseSubjectsSchedule();
349     break;
350
351 case 'check_schedule_conflict':
352     require_once BASE_PATH . '/controllers/ScheduleController.php';
353     $controller = new ScheduleController();
354     $controller->checkScheduleConflict();
355     break;
356
357 case 'institution':
358     require_once BASE_PATH . '/controllers/InstitutionController.php';
359     $controller = new InstitutionController();
360     $controller->index();
361     break;
362
363 case 'update_institution':
364     require_once BASE_PATH . '/controllers/InstitutionController.php';
365     $controller = new InstitutionController();
366     $controller->update();
367     break;
368
369 case 'assign_institution_shift':
370     require_once BASE_PATH . '/controllers/InstitutionController.php';
371     $controller = new InstitutionController();
372     $controller->assignShift();
373     break;
374
375 case 'remove_institution_shift':
376     require_once BASE_PATH . '/controllers/InstitutionController.php';
377     $controller = new InstitutionController();
378     $controller->removeShift();
379     break;
380
381 case 'create_user':
382     require_once BASE_PATH . '/controllers/UserController.php';
383     $controller = new UserController();
384     $controller->create();
385     break;
386
387 case 'edit_user':
388     require_once BASE_PATH . '/controllers/UserController.php';
389     $controller = new UserController();
390     $controller->edit();
391     break;
```

```

392     case 'delete_user':
393         require_once BASE_PATH . '/controllers/UserController.php';
394         $controller = new UserController();
395         $controller->delete();
396         break;
397
398     case 'create_school_year':
399         require_once BASE_PATH . '/controllers/AcademicController.php';
400         $controller = new AcademicController();
401
402         $controller->createSchoolYear();
403         break;
404
405     case 'edit_school_year':
406         require_once BASE_PATH . '/controllers/AcademicController.php';
407         $controller = new AcademicController();
408         $controller->editSchoolYear();
409         break;
410
411     case 'delete_school_year':
412         require_once BASE_PATH . '/controllers/AcademicController.php';
413         $controller = new AcademicController();
414         $controller->deleteSchoolYear();
415         break;
416
417     case 'activate_school_year':
418         require_once BASE_PATH . '/controllers/AcademicController.php';
419         $controller = new AcademicController();
420         $controller->activateSchoolYear();
421         break;
422
423     case 'deactivate_school_year':
424         require_once BASE_PATH . '/controllers/AcademicController.php';
425         $controller = new AcademicController();
426         $controller->deactivateSchoolYear();
427         break;
428
429     case 'edit_course':
430         require_once BASE_PATH . '/controllers/AcademicController.php';
431         $controller = new AcademicController();
432         $controller->editCourse();
433         break;
434
435     case 'delete_course':
436         require_once BASE_PATH . '/controllers/AcademicController.php';
437         $controller = new AcademicController();
438         $controller->deleteCourse();
439         break;
440
441     case 'edit_subject':
442         require_once BASE_PATH . '/controllers/AcademicController.php';
443         $controller = new AcademicController();
444         $controller->editSubject();
445         break;
446
447     case 'delete_subject':
448         require_once BASE_PATH . '/controllers/AcademicController.php';
449         $controller = new AcademicController();
450         $controller->deleteSubject();
451         break;
452
453     case 'unenroll_student':
454         require_once BASE_PATH . '/controllers/AcademicController.php';
455         $controller = new AcademicController();
456         $controller->unenrollStudent();
457         break;
458
459     case 'delete_backup':
460         require_once BASE_PATH . '/controllers/BackupController.php';
461         $controller = new BackupController();
462         $controller->delete();
463         break;
464
465     case 'remove_representative':
466         require_once BASE_PATH . '/controllers/RepresentativeController.php';
467         $controller = new RepresentativeController();
468         $controller->removeRelation();
469         break;
470
471     case 'notifications':

```

```
471         require_once BASE_PATH . '/controllers/NotificationController.php';
472         (new NotificationController())->index();
473         break;
474
475     case 'notifications_mark_read':
476         require_once BASE_PATH . '/controllers/NotificationController.php';
477         (new NotificationController())->markRead();
478         break;
479
480     case 'notifications_mark_all':
481
482         require_once BASE_PATH . '/controllers/NotificationController.php';
483         (new NotificationController())->markAllRead();
484         break;
485
486     case 'notifications_delete':
487         require_once BASE_PATH . '/controllers/NotificationController.php';
488         (new NotificationController())->delete();
489         break;
490
491     case 'notifications_delete_read':
492         require_once BASE_PATH . '/controllers/NotificationController.php';
493         (new NotificationController())->deleteRead();
494         break;
495
496     case 'notifications_unread_json':
497         require_once BASE_PATH . '/controllers/NotificationController.php';
498         (new NotificationController())->getUnread();
499         break;
500
501     case 'tutor_course_attendance':
502         require_once BASE_PATH . '/controllers/TutorController.php';
503         (new TutorController())->courseAttendance();
504         break;
505
506     case 'tutor_course_attendance_ajax':
507         require_once BASE_PATH . '/controllers/TutorController.php';
508         (new TutorController())->ajax();
509         break;
510
511     default:
512         header('Location: ?action=login');
```

## 43. course\_edit.php

Ruta: 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 ...
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: ...
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>g...
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)&l...
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'] == $shift['id'] ? 'selected' : '' ?>&gt;
124                 <?=$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
162         'Administración': ['Asistencia Administrativa', 'Gestión Empresarial'],
163         'Contabilidad': ['Contabilidad', 'Ventas e Información Comercial'],
164         'Comercialización y Ventas': ['Ventas e Información Comercial', 'Marketing'],
165         'Servicios Hoteleros': ['Hotelería', 'Hospitalidad'],
166         'Turismo': ['Turismo', 'Guía Turístico'],
167         'Electromecánica Automotriz': ['Electromecánica Automotriz', 'Mecánica Automotriz'],
168         'Instalaciones Eléctricas': ['Electricidad', 'Instalaciones Eléctricas'],
169         'Electrónica de Consumo': ['Electrónica', 'Mantenimiento Electrónico'],
170         'Atención Integral en Salud': ['Atención en Enfermería', 'Auxiliar de Salud'],
171         'Producción Agropecuaria': ['Producción Agropecuaria', 'Agroindustria'],
172         'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
173         'Diseño Gráfico': ['Diseño Gráfico', 'Multimedia'],
174         'Servicios de Belleza': ['Peluquería', 'Cosmetología']
175     };
176
177     // Detectar tipo educativo desde el grade_level actual
178     function detectType(grade) {
179         if (!grade) return '';
180         if (grade.includes('Inicial')) return 'inicial';
181         if (grade.includes('EGB')) return 'egb';
182         if (grade.includes('BGU')) return 'bgu';
183         if (grade.includes('BT')) return 'bt';
184         return '';
185     }
186
187     function getNocturnaOption() {
188         const opts = document.getElementById('shift_id').options;
189         for (let i = 0; i < opts.length; i++) {
190             if ((opts[i].getAttribute('data-shift') || '').toLowerCase() === 'nocturna') return opts[i];
191         }
192         return null;
193     }
194
195     function updateNocturnaVisibility(type, grade) {
196         const nocturna = getNocturnaOption();
197         if (!nocturna) return;
198         let visible = (type === 'bgu' || type === 'bt' || (type === 'egb' && egbNocturnaAllowed.includes(grade)))...
199         nocturna.style.display = visible ? '' : 'none';
200         if (!visible && nocturna.selected) {
201             document.getElementById('shift_id').value = '';
202         }
203     }
204
205     function updateGradeLevels(preselectGrade) {
206         const type = document.getElementById('education_type').value;
207         const gradeSelect = document.getElementById('grade_level');
208         gradeSelect.innerHTML = '<option value="">Seleccionar grado...</option>';
209
210         if (!type) return;
211
212         (gradeLevels[type] || []).forEach(function(g) {
213             const opt = document.createElement('option');
214             opt.value = g;
215             opt.textContent = g;
216             if (preselectGrade && g === preselectGrade) opt.selected = true;
217             gradeSelect.appendChild(opt);
218         });
219
220         const isBT = (type === 'bt');
221         document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
222         document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
223         document.getElementById('specialty').required = isBT;
224
225         if (!isBT) {
226             document.getElementById('specialty').value = '';
227             document.getElementById('carrera').innerHTML = '<option value="">Sin especificar</option>';
228         }
229
230         updateNocturnaVisibility(type, gradeSelect.value);
231     }
232
233     function onGradeChange() {
234         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
242     const carreraSelect = document.getElementById('carrera');
243     carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
244     if (carreras[fig]) {
245         carreras[fig].forEach(function(c) {
246             const opt = document.createElement('option');
247             opt.value = c; opt.textContent = c;
248             carreraSelect.appendChild(opt);
249         });
250     }
251     generateCourseName();
252 }
253
254 function generateCourseName() {
255     const grade = document.getElementById('grade_level').value;
256     const parallel = document.getElementById('parallel').value;
257     const shiftSelect = document.getElementById('shift_id');
258     const shiftOpt = shiftSelect.options[shiftSelect.selectedIndex];
259     const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';
260     const type = document.getElementById('education_type').value;
261     const specialty = document.getElementById('specialty').value;
262     const carrera = document.getElementById('carrera').value;
263
264     if (!grade || !parallel || !shiftName) return;
265
266     let name = grade + ' ' + parallel + ' ';
267     if (type === 'bt' && specialty) {
268         name += ' - ' + specialty;
269         if (carrera) name += ' (' + carrera + ')';
270     }
271     name += ' - ' + shiftName.charAt(0).toUpperCase() + shiftName.slice(1);
272     document.getElementById('course_name').value = name;
273 }
274
275 // Extraer figura profesional y carrera desde el nombre del curso
276 // Formato: "1.º BT "A" - Informática (Soporte Técnico) - Matutina"
277 function parseSpecialtyFromName(courseName) {
278     const result = { specialty: '', carrera: '' };
279     // Buscar patrón: - Figura (Carrera) - o - Figura -
280     const match = courseName.match(/BT "[A-Z]" - ([^()]+)(?:\s*\([^\)]+\))?\s*-\s*\w+$/);
281     if (match) {
282         result.specialty = match[1].trim();
283         result.carrera = match[2] ? match[2].trim() : '';
284     }
285     return result;
286 }
287
288 // PRECARGAR datos del curso al cargar la página
289 window.addEventListener('DOMContentLoaded', function() {
290     const detectedType = detectType(currentGradeLevel);
291     if (detectedType) {
292         document.getElementById('education_type').value = detectedType;
293         updateGradeLevels(currentGradeLevel);
294     }
295
296     // Preseleccionar jornada
297     const shiftSelect = document.getElementById('shift_id');
298     for (let i = 0; i < shiftSelect.options.length; i++) {
299         if (shiftSelect.options[i].value === currentShiftId) {
300             shiftSelect.selectedIndex = i;
301             break;
302         }
303     }
304
305     // Precargar figura profesional y carrera (solo BT)
306     if (detectedType === 'bt') {
307         const parsed = parseSpecialtyFromName(currentGradeLevel + ' - ' + &lt;? json_encode($course['name']) ?>);
308
309         // Intentar parsear directo del nombre completo del curso
310         const fullName = &lt;? json_encode($course['name']) ?>;
311         const matchFull = fullName.match(/- ([^()]+)(?:\s*\([^\)]+\))?\s*-\s*\w+$/);
312         if (matchFull) {
313             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 = '<option value="">Sin especificar</option>';
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 </script>
341 </body>
342 </html>

```

## 44. course\_students.php

Ruta: 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      <title>Estudiantes 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); }
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         h2 { margin-bottom: 10px; color: #333; }
19         .course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
20     </style>
21 </head>
22 <body>
23     <div class="navbar">
24         <h1>Estudiantes del Curso</h1>
25     </div>
26     <a href="?action=academic">← Configuración Académica</a>
27     <a href="?action=logout">Cerrar sesión</a>
28 </div>
29 </div>
30
31 <div class="container">
32     <div class="card">
33         <div class="course-info">
34             <h2><?=$course['name'] ?></h2>
35             <p><strong>Nivel:</strong> <?=$course['grade_level'] ?> | <strong>Paralel...
36             <p><strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?> | <strong>...
37         </div>
38
39         <h3>Estudiantes Matriculados (<?= count($students) ?>)</h3>
40
41         <?php if(count($students) > 0): ?>
42             <table>
43                 <thead>
44                     <tr>
45                         <th>#</th>
46                         <th>Apellidos y Nombres</th>
47                         <th>Cédula</th>
48                         <th>Fecha Matrícula</th>
49                     </tr>
50                 </thead>
51                 <tbody>
52                     <?php foreach($students as $index => $student): ?>
53                         <tr>
54                             <td><?=$index + 1 ?></td>
55                             <td><?=$student['last_name'] . ' ' . $student['first_name'] ?></td>
56                             <td><?=$student['dni'] ?? '-' ?></td>
57                             <td><?=$student['enrollment_date'] ?></td>
58                         </tr>
59                     <?php endforeach; ?>
60                 </tbody>
61             </table>
62             <?php else: ?>
63                 <p style="color: #666; margin-top: 20px;">No hay estudiantes matriculados en este curso.</p>
64             <?php endif; ?>
65         </div>
66     </div>
67 </body>
68 </html>
```

## 45. create.php

Ruta: ArchImp/views/academic/create.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>Crear 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     </style>
40 </head>
41 <body>
42     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
43
44     <div class="container">
45         <div class="row justify-content-center">
46             <div class="col-md-8">
47                 <div class="card shadow">
48                     <div class="card-header bg-primary text-white">
49                         <h4 class="mb-0">■ Crear Nuevo Año Lectivo</h4>
50                     </div>
51                     <div class="card-body">
52                         <?php if (!empty($errors)): ?>
53                             <div class="alert alert-danger">
54                                 <strong>Errores:</strong>
55                                 <ul class="mb-0">
56                                     <?php foreach ($errors as $error): ?>
57                                         <li>&#x2022; <= htmlspecialchars($error) ?></li>
58                                     <?php endforeach; ?>
59                                 </ul>
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
82                 placeholder="Ej: 2025-2026"
83                 value="<?php if (isset($_POST['name'])) { echo htmlspecialchars($_POST['name']); } else { echo '...'; } >"
84                 required>
85             <small class="text-muted">Formato recomendado: YYYY-YYYY</small>
86         </div>
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="<?php if (isset($_POST['start_date'])) { echo $_POST['start_date']; } else { echo '...'; } >"
94                 required>
95         </div>
96         <div class="col-md-6 mb-3">
97             <label class="form-label required">Fecha de Fin</label>
98             <input type="date"
99                 name="end_date"
100                 class="form-control"
101                 value="<?php if (isset($_POST['end_date'])) { echo $_POST['end_date']; } else { echo '...'; } >"
102                 required>
103         </div>
104     </div>
105
106     <div class="form-check">
107         <input class="form-check-input"
108             type="checkbox"
109             name="is_active"
110             id="is_active"
111             <?php if (isset($_POST['is_active'])) { echo '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>

```

## 46. edit.php

**Ruta:** 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: <?php echo htmlspecialchars($year['name']); ?>&l...
66                             <span class="status-badge <?php echo ($year['is_active'] ? 'status-active' : 'status-inactive'...
67                                 <?php echo ($year['is_active'] ? '✓ ACTIVO' : 'Inactivo'); ?>
68                         </span>
69                     </div>
70                 </div>
71                 <div class="card-body">
72                     <?php if (!empty($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>

```



## 47. enroll.php

Ruta: 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        .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-bot...
15        .form-group { margin-bottom: 20px; }
16        label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
17        select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
18        .student-list { margin-top: 20px; }
19        .student-item { padding: 10px; border: 1px solid #ddd; border-radius: 4px; margin-bottom: 10px; display: flex; a...
20        .student-item input[type="checkbox"] { margin-right: 10px; }
21        button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointe...
22        button:hover { background: #218838; }
23        .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
24        .warning { background: #fff3cd; color: #856404; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
25        .info { background: #d1ecf1; color: #0c5460; padding: 10px; border-radius: 4px; margin-bottom: 15px; }
26        table { width: 100%; border-collapse: collapse; margin-top: 15px; }
27        th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; }
28        th { background: #f8f9fa; font-weight: bold; }
29        .badge { display: inline-block; padding: 4px 8px; border-radius: 3px; font-size: 12px; background: #007bff; colo...
30    </style>
31 </head>
32 <body>
33     <div class="navbar">
34         <h1>Matricular Estudiantes</h1>
35     </div>
36     <a href="?action=academic">← Configuración Académica</a>
37     <a href="?action=logout">Cerrar sesión</a>
38 </div>
39 </div>
40
41 <div class="container">
42     <?php if(isset($_GET['enrolled'])): ?>
43         <div class="success">✓ <?=$_GET['enrolled'] ?> estudiante(s) matriculado(s) correctamente</div>
44     <?php endif; ?>
45
46     <?php if(isset($_GET['errors']) && $_GET['errors'] > 0): ?>
47         <div class="warning">■ <?=$_GET['errors'] ?> estudiante(s) no pudieron ser matriculados (ya est...
48     <?php endif; ?>
49
50     <?php if(isset($_GET['unenrolled'])): ?>
51         <div class="success">✓ Estudiante retirado del curso correctamente</div>
52     <?php endif; ?>
53
54     <?php if(isset($_GET['error']) && $_GET['error'] === 'not_enrolled'): ?>
55         <div class="warning">■ El estudiante no está matriculado en ningún curso</div>
56     <?php endif; ?>
57
58     <?php if(isset($_GET['error']) && $_GET['error'] === 'unenroll_failed'): ?>
59         <div class="warning">✗ Error al retirar el estudiante del curso</div>
60     <?php endif; ?>
61
62     <!-- Resumen de estudiantes -->
63     <div class="card">
64         <h2>Resumen</h2>
65         <div class="info">
66             <strong>Total estudiantes registrados:</strong> <?=$_count($allStudents) ?><br>
67             <strong>Ya matriculados:</strong> <?=$_count($allStudents) - count($availableStudents) ?>...
68             <strong>Disponibles para matricular:</strong> <?=$_count($availableStudents) ?>
69         </div>
70     </div>
71
72     <!-- Formulario de matrícula -->
73     <?php if(count($availableStudents) > 0): ?>
74         <div class="card">
75             <h2>Matricular Estudiantes Disponibles</h2>
```

```

76     <p style="color: #666; margin-bottom: 20px;">
77         Solo se muestran estudiantes que aún NO están matriculados en ningún curso del año lectivo actual.
78     </p>
79
80     <form method="POST">
81
82         <div class="form-group">
83             <label>Curso</label>
84             <select name="course_id" required>
85                 <option value="">Seleccionar curso...</option>
86                 <?php foreach($courses as $course): ?>
87                     <option value="<? $course['id'] ?>"><? $course['name'] ?> - <? $course['shift_name'] ?>
88                 </option>
89                 <?php endforeach; ?>
90             </select>
91         </div>
92
93         <div class="student-list">
94             <label>Seleccionar Estudiantes</label>
95             <?php foreach($availableStudents as $student): ?>
96                 <div class="student-item">
97                     <input type="checkbox" name="student_ids[]" value="<? $student['id'] ?>" id="student_&...
98                     <label for="student_<? $student['id'] ?>" style="margin: 0; font-weight: normal;">
99                         <? $student['last_name'] . ' ' . $student['first_name'] ?>
100                         (<? $student['dni'] ?? 'Sin cédula' ?>)
101                     </label>
102                 </div>
103             <?php endforeach; ?>
104         </div>
105
106         <button type="submit">Matricular Seleccionados</button>
107     </form>
108 </div>
109 <?php else: ?>
110 <div class="card">
111     <div class="warning">
112         ■ No hay estudiantes disponibles para matricular. Todos los estudiantes ya están asignados a cursos.
113     </div>
114 </div>
115 <?php endif; ?>
116
117 <!-- Lista de todos los estudiantes con su estado -->
118 <div class="card">
119     <h2>Estado de Todos los Estudiantes</h2>
120     <table>
121         <thead>
122             <tr>
123                 <th>#</th>
124                 <th>Apellidos y Nombres</th>
125                 <th>Cédula</th>
126                 <th>Curso</th>
127                 <th>Estado</th>
128                 <th>Acciones</th>
129             </tr>
130         </thead>
131         <tbody>
132             <?php
133                 $counter = 1;
134                 foreach($allStudents as $student):
135                     $course = $this->userModel->getStudentCourse($student['id'], $activeYear['id']);
136                     <?>
137                     <tr>
138                         <td><? $counter++ ?></td>
139                         <td><? $student['last_name'] . ' ' . $student['first_name'] ?></td>
140                         <td><? $student['dni'] ?? '-' ?></td>
141                         <td>
142                             <?php if($course): ?>
143                                 <strong><? $course['name'] ?></strong>
144                                 <br>
145                                 <small style="color: #666;"><? $course['shift_name'] ?></small>
146                             <?php else: ?>
147                                 <span style="color: #999;">-</span>
148                             <?php endif; ?>
149                         </td>
150                         <td>
151                             <?php if($course): ?>
152                                 <span class="badge" style="background: #28a745;">
153                                     ✓ Matriculado
154                                 </span>

```

```

155         <?php else: ?>
156             <span class="badge" style="background: #6c757d;">Sin matricular</span>
157         <?php endif: ?>
158     </td>
159 </td>
160     <?php if($course): ?>
161         <form method="POST" action="?action=unenroll_student" style="display: inline;"
162             onsubmit="return confirmUnenroll(event, '= addslashes($student['last_name']) ...
163             &lt;input type="hidden" name="student_id" value="<?= $student['id'] ?&gt;"&gt;
164             &lt;button type="submit"
165                 style="padding: 6px 12px; background: #dc3545; font-size: 13px;"&gt;
166                 X Retirar
167             &lt;/button&gt;
168         &lt;/form&gt;
169     &lt;?php else: ?&gt;
170         &lt;span style="color: #999;"&gt;-&lt;/span&gt;
171     &lt;?php endif: ?&gt;
172 &lt;/td&gt;
173 &lt;/tr&gt;
174 &lt;?php endforeach: ?&gt;
175 &lt;/tbody&gt;
176 &lt;/table&gt;
177 &lt;/div&gt;
178 &lt;/div&gt;
179
180 &lt;script&gt;
181 function confirmUnenroll(event, studentName, courseName) {
182     event.preventDefault();
183
184     const modal = document.createElement('div');
185     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
186
187     const modalContent = document.createElement('div');
188     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
189
190     modalContent.innerHTML = `
191         &lt;h3 style="margin: 0 0 15px 0; color: #dc3545;"&gt;Retirar Estudiante del Curso&lt;/h3&gt;
192         &lt;p style="margin: 0 0 20px 0; color: #666;"&gt;
193             ¿Está seguro de retirar a &lt;strong&gt;${studentName}&lt;/strong&gt; del curso &lt;strong&gt;${courseName}&lt;/strong&gt;?
194         &lt;/p&gt;
195         &lt;p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #fff3cd; padding: 10px; border-ra...
196             &lt;strong&gt;Advertencia&lt;/strong&gt; Esta acción quitará al estudiante del curso.
197             Los registros de asistencia se conservarán pero el estudiante no aparecerá como matriculado.
198         &lt;/p&gt;
199         &lt;div style="display: flex; gap: 10px; justify-content: flex-end;"&gt;
200             &lt;button type="button" id="cancelUnenrollBtn" style="padding: 10px 20px; background: #6c757d; color: w...
201                 Cancelar
202             &lt;/button&gt;
203             &lt;button type="button" id="confirmUnenrollBtn" style="padding: 10px 20px; background: #dc3545; color: ...
204                 Sí, Retirar
205             &lt;/button&gt;
206         &lt;/div&gt;
207     `;
208
209     modal.appendChild(modalContent);
210     document.body.appendChild(modal);
211
212     const form = event.target;
213
214     document.getElementById('confirmUnenrollBtn').onclick = function() {
215         document.body.removeChild(modal);
216         form.submit();
217     };
218
219     document.getElementById('cancelUnenrollBtn').onclick = function() {
220         document.body.removeChild(modal);
221     };
222
223     return false;
224 }
225 &lt;/script&gt;
226 &lt;/body&gt;
227 &lt;/html&gt;
</pre

```

## 48. index.php

Ruta: 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: 20...
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: pointe...
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
34     <div class="container">
35         <?php if(isset($_GET['course_success'])): ?>
36             <div class="success">✓ Curso creado correctamente</div>
37         <?php endif; ?>
38         <?php if(isset($_GET['course_updated'])): ?>
39             <div class="success">✓ Curso actualizado correctamente</div>
40         <?php endif; ?>
41         <?php if(isset($_GET['course_deleted'])): ?>
42             <div class="success">✓ Curso eliminado correctamente</div>
43         <?php endif; ?>
44
45         <?php if(isset($_GET['subject_success'])): ?>
46             <div class="success">✓ Asignatura creada correctamente</div>
47         <?php endif; ?>
48         <?php if(isset($_GET['subject_updated'])): ?>
49             <div class="success">✓ Asignatura actualizada correctamente</div>
50         <?php endif; ?>
51         <?php if(isset($_GET['subject_deleted'])): ?>
52             <div class="success">✓ Asignatura eliminada correctamente</div>
53         <?php endif; ?>
54
55         <?php if(isset($_GET['sy_created'])): ?>
56             <div class="success">✓ Año lectivo creado correctamente</div>
57         <?php endif; ?>
58         <?php if(isset($_GET['sy_updated'])): ?>
59             <div class="success">✓ Año lectivo actualizado correctamente</div>
60         <?php endif; ?>
61         <?php if(isset($_GET['sy_deleted'])): ?>
62             <div class="success">✓ Año lectivo eliminado correctamente</div>
63         <?php endif; ?>
64         <?php if(isset($_GET['sy_activated'])): ?>
65             <div class="success">✓ Año lectivo activado correctamente</div>
66         <?php endif; ?>
67         <?php if(isset($_GET['sy_deactivated'])): ?>
68             <div class="success">✓ Año lectivo desactivado correctamente</div>
69         <?php endif; ?>
70
71         <?php if(isset($_GET['error']) && $_GET['error'] === 'no_active_year'): ?>
72             <div class="error">✗ No hay un año lectivo activo</div>
73         <?php endif; ?>
74         <?php if(isset($_GET['error']) && $_GET['error'] === 'course_not_found'): ?>
75             <div class="error">✗ Curso no encontrado</div>
76         <?php endif; ?>
77     </div>
```

```

76     <?php endif; ?>;
77     <?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_students'): ?>;
78         <div class="error"><X No se puede eliminar el curso porque tiene estudiantes matriculados</div>;
79     <?php endif; ?>;
80     <?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_assignments'): ?>;
81         <div class="error"><X No se puede eliminar el curso porque tiene asignaciones docentes</div>;
82     <?php endif; ?>;
83     <?php if(isset($_GET['error']) && $_GET['error'] === 'subject_not_found'): ?>;
84         <div class="error"><X Asignatura no encontrada</div>;
85     <?php endif; ?>;
86     <?php if(isset($_GET['error']) && $_GET['error'] === 'subject_has_assignments'): ?>;
87         <div class="error"><X No se puede eliminar la asignatura porque tiene asignaciones docentes</div>;
88     <?php endif; ?>;
89     <?php if(isset($_GET['error']) && $_GET['error'] === 'cannot_delete_active'): ?>;
90         <div class="error"><X No se puede eliminar el año lectivo activo</div>;
91     <?php endif; ?>;
92     <?php if(isset($_GET['error']) && $_GET['error'] === 'has_courses'): ?>;
93         <div class="error"><X No se puede eliminar el año lectivo porque tiene cursos asociados</div>;
94     <?php endif; ?>;
95     <?php if(isset($_GET['error']) && $_GET['error'] === 'year_not_found'): ?>;
96         <div class="error"><X Año lectivo no encontrado</div>;
97     <?php endif; ?>;
98
99     <!-- Años Lectivos -->;
100    <div class="card" style="margin-bottom: 20px;">;
101        <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px;">;
102            <h2 style="margin: 0;">■ Años Lectivos</h2>;
103            <button onclick="location.href='?action=create_school_year'" class="btn-primary" style="margin: 0;">...
104                + Crear Año Lectivo
105            </button>;
106        </div>;
107
108        <table>;
109            <thead>;
110                <tr>;
111                    <th>#</th>;
112                    <th>Nombre</th>;
113                    <th>Fecha Inicio</th>;
114                    <th>Fecha Fin</th>;
115                    <th>Estado</th>;
116                    <th>Acciones</th>;
117                </tr>;
118            </thead>;
119            <tbody>;
120                <?php if(empty($schoolYears)): ?>;
121                    <tr>;
122                        <td colspan="6" style="text-align:center; color: #999;">No hay años lectivos registrad...
123                    </tr>;
124                <?php else: ?>;
125                    <?php $counter = 1; foreach($schoolYears as $year): ?>;
126                        <tr style="background: #e7f3ff;">
127                            <td>#<?php $counter++ ?></td>;
128                            <td><strong><?php htmlspecialchars($year['name']) ?></strong></td>;
129                            <td><?php date('d/m/Y', strtotime($year['start_date'])) ?></td>;
130                            <td><?php date('d/m/Y', strtotime($year['end_date'])) ?></td>;
131                            <td><?php if($year['is_active']): ?>;
132                                <span style="background: #28a745; color: white; padding: 4px 10px; border-radius:...
133                                    ✓ ACTIVO
134                                </span>;
135                            <?php else: ?>;
136                                <span style="background: #6c757d; color: white; padding: 4px 10px; border-radius:...
137                                    Inactivo
138                                </span>;
139                            <?php endif; ?>;
140                        </td>;
141                    </tr>;
142                    <td style="white-space: nowrap;">;
143                        <button onclick="location.href='?action=edit_school_year&id=<?php $year['id'] ?>...'
144                            style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #212529;">...
145                            ■ Editar
146                        </button>;
147
148                    <?php if($year['is_active']): ?>;
149                        <form method="POST" action="?action=deactivate_school_year" style="display: inlin...
150                            onsubmit="return confirm('¿Desactivar este año lectivo?')">;
151                            <input type="hidden" name="year_id" value="<?php $year['id'] ?>">;
152                            <button type="submit" style="padding: 5px 10px; font-size: 12px; background: ...
153                                ■ Desactivar
154                            </button>;

```

```

155         </form>;
156         <?php else: ?>
157             <form method="POST" action="?action=activate_school_year" style="display: inline;...
158                 onsubmit="return confirm('¿Activar este año lectivo? Se desactivarán los demás...
159                 <input type="hidden" name="year_id" value="<?= $year['id'] ?>";>
160                 <button type="submit" style="padding: 5px 10px; font-size: 12px; background: ...

161             ✓ Activar
162             </button>;
163         </form>;
164
165         <form method="POST" action="?action=delete_school_year" style="display: inline;"
166             onsubmit="return confirmDelete(event, '<?= htmlspecialchars($year['name']) ...
167             <input type="hidden" name="year_id" value="<?= $year['id'] ?>";>
168             <button type="submit" style="padding: 5px 10px; font-size: 12px; background: ...
169             ■■ Eliminar
170             </button>;
171         </form>;
172         <?php endif; ?>
173     </td>;
174 </tr>;
175 <?php endforeach; ?>
176 <?php endif; ?>
177 </tbody>;
178 </table>;
179 </div>;
180
181 <div class="grid">;
182     <!-- Crear Curso -->;
183     <div class="card">;
184         <h2>Crear Curso</h2>;
185         <form method="POST" action="?action=create_course" id="courseForm">;
186
187         <!-- NIVEL EDUCATIVO -->;
188         <div class="form-group">;
189             <label>Nivel Educativo</label>;
190             <select name="education_type" id="education_type" required onchange="updateGradeLevels()">;
191                 <option value="">Seleccionar tipo...</option>;
192                 <option value="inicial">■ Educación Inicial</option>;
193                 <option value="egb">■ Educación General Básica (EGB)</option>;
194                 <option value="bgu">■ Bachillerato General Unificado (BGU)</option>;
195                 <option value="bt">■ Bachillerato Técnico (BT)</option>;
196             </select>;
197         </div>;
198
199         <!-- GRADO -->;
200         <div class="form-group" id="group_grade">;
201             <label>Grado / Año</label>;
202             <select name="grade_level" id="grade_level" required onchange="onGradeChange()">;
203                 <option value="">Seleccione nivel primero...</option>;
204             </select>;
205         </div>;
206
207         <!-- FIGURA PROFESIONAL (solo BT) -->;
208         <div class="form-group" id="group_specialty" style="display:none;">;
209             <label>Figura Profesional</label>;
210             <select name="specialty" id="specialty" onchange="updateCarreras()">;
211                 <option value="">Seleccionar figura...</option>;
212                 <option value="Informática">Informática</option>;
213                 <option value="Administración">Administración</option>;
214                 <option value="Contabilidad">Contabilidad</option>;
215                 <option value="Comercialización y Ventas">Comercialización y Ventas</option>;
216                 <option value="Servicios Hoteleros">Servicios Hoteleros</option>;
217                 <option value="Turismo">Turismo</option>;
218                 <option value="Electromecánica Automotriz">Electromecánica Automotriz</option>;
219                 <option value="Instalaciones Eléctricas">Instalaciones Eléctricas</option>;
220                 <option value="Electrónica de Consumo">Electrónica de Consumo</option>;
221                 <option value="Mecanizado y Construcciones Metálicas">Mecanizado y Construcciones Metá...
222                 <option value="Producción Agropecuaria">Producción Agropecuaria</option>;
223                 <option value="Acuicultura">Acuicultura</option>;
224                 <option value="Industrialización de Alimentos">Industrialización de Alimentos</opti...
225                 <option value="Confección Textil">Confección Textil</option>;
226                 <option value="Música">Música</option>;
227                 <option value="Artes Plásticas">Artes Plásticas</option>;
228                 <option value="Diseño Gráfico">Diseño Gráfico</option>;
229                 <option value="Servicios de Belleza">Servicios de Belleza</option>;
230                 <option value="Atención Integral en Salud">Atención Integral en Salud</option>;
231                 <option value="Mecatrónica">Mecatrónica</option>;
232                 <option value="Refrigeración y Climatización">Refrigeración y Climatización</option...
233                 <option value="Construcción">Construcción</option>;

```

```

234         <option value="Redes y Telecomunicaciones">Redes y Telecomunicaciones</option>;
235         <option value="Seguridad Industrial">Seguridad Industrial</option>;
236         <option value="Logística y Transporte">Logística y Transporte</option>;
237     </select>;
238 </div>;
239
240 <!-- ESPECIALIDAD / CARRERA (solo BT) -->;
241
242 <div class="form-group" id="group_carrera" style="display:none">;
243     <label>Especialidad / Carrera <span style="color:#999; font-weight:normal;">(opciona...
244     <select name="carrera" id="carrera" onchange="generateCourseName()">;
245         <option value="">Sin especificar</option>;
246     </select>;
247 </div>;
248
249 <!-- PARALELO -->;
250 <div class="form-group">;
251     <label>Paralelo</label>;
252     <select name="parallel" id="parallel" required onchange="generateCourseName()">;
253         <option value="">Seleccionar...</option>;
254         <?php foreach(range('A', 'J') as $letter): ?>
255             <option value="<?=$letter ?>"><?=$letter ?></option>;
256         <?php endforeach; ?>
257     </select>;
258 </div>;
259
260 <!-- JORNADA -->;
261 <div class="form-group">;
262     <label>Jornada</label>;
263     <select name="shift_id" id="shift_id" required onchange="generateCourseName()">;
264         <option value="">Seleccionar...</option>;
265         <?php foreach($shifts as $shift): ?>
266             <option value="<?=$shift['id'] ?>" data-shift="<?=$shift['name'] ?>"><?=$shift['name'] ?></option>;
267         <?php endforeach; ?>
268     </select>;
269 </div>;
270
271 <!-- NOMBRE GENERADO -->;
272 <div class="form-group">;
273     <label>Nombre del Curso <span style="color:#999; font-weight:normal;">(generado auto...
274     <input type="text" name="name" id="course_name" readonly
275         style="background: #f0f0f0; font-weight: bold; color: #333;">;
276 </div>;
277
278 <button type="submit">Crear Curso</button>;
279 </form>;
280 </div>;
281
282
283 <script>;
284 const gradeLevels = {
285     inicial: [
286         'Inicial 1 (0-3 años)',
287         'Inicial 2 (3-5 años)'
288     ],
289     egb: [
290         '1.º EGB - Preparatoria',
291         '2.º EGB', '3.º EGB', '4.º EGB',
292         '5.º EGB', '6.º EGB', '7.º EGB',
293         '8.º EGB', '9.º EGB', '10.º EGB'
294     ],
295     bgu: ['1.º BGU', '2.º BGU', '3.º BGU'],
296     bt: ['1.º BT', '2.º BT', '3.º BT']
297 };
298
299 // Grados EGB donde se permite jornada nocturna
300 const egbNocturnaAllowed = ['8.º EGB', '9.º EGB', '10.º EGB'];
301
302 const carreras = {
303     'Informática': ['Aplicaciones Informáticas', 'Programación de Software', 'Soporte Técnico', 'Sistemas Micro...'],
304     'Administración': ['Asistencia Administrativa', 'Gestión Empresarial'],
305     'Contabilidad': ['Contabilidad', 'Ventas e Información Comercial'],
306     'Comercialización y Ventas': ['Ventas e Información Comercial', 'Marketing'],
307     'Servicios Hoteleros': ['Hotelería', 'Hospitalidad'],
308     'Turismo': ['Turismo', 'Guía Turístico'],
309     'Electromecánica Automotriz': ['Electromecánica Automotriz', 'Mecánica Automotriz'],
310     'Instalaciones Eléctricas': ['Electricidad', 'Instalaciones Eléctricas'],
311     'Electrónica de Consumo': ['Electrónica', 'Mantenimiento Electrónico'],
312     'Atención Integral en Salud': ['Atención en Enfermería', 'Auxiliar de Salud'],

```

```

313     'Producción Agropecuaria': ['Producción Agropecuaria', 'Agroindustria'],
314     'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
315     'Diseño Gráfico': ['Diseño Gráfico', 'Multimedia'],
316     'Servicios de Belleza': ['Peluquería', 'Cosmetología']
317 };
318
319 function getNocturnaOption() {
320     const opts = document.getElementById('shift_id').options;
321
322     for (let i = 0; i < opts.length; i++) {
323         if ((opts[i].getAttribute('data-shift') || '').toLowerCase() === 'nocturna') return opts[i];
324     }
325     return null;
326 }
327
328 function updateNocturnaVisibility(type, grade) {
329     const nocturna = getNocturnaOption();
330     if (!nocturna) return;
331
332     let visible = false;
333     if (type === 'bgu' || type === 'bt') {
334         visible = true;
335     } else if (type === 'egb' && egbNocturnaAllowed.includes(grade)) {
336         visible = true;
337     }
338
339     nocturna.style.display = visible ? '' : 'none';
340
341     // Si estaba seleccionada y ya no aplica, limpiar
342     if (!visible && nocturna.selected) {
343         document.getElementById('shift_id').value = '';
344         generateCourseName();
345     }
346 }
347
348 function updateGradeLevels() {
349     const type = document.getElementById('education_type').value;
350     const gradeSelect = document.getElementById('grade_level');
351
352     gradeSelect.innerHTML = '<option value="">Seleccionar grado...</option>';
353     document.getElementById('course_name').value = '';
354
355     if (!type) return;
356
357     (gradeLevels[type] || []).forEach(function(g) {
358         const opt = document.createElement('option');
359         opt.value = g;
360         opt.textContent = g;
361         gradeSelect.appendChild(opt);
362     });
363
364     // Mostrar/ocultar campos BT
365     const isBT = (type === 'bt');
366     document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
367     document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
368     document.getElementById('specialty').required = isBT;
369
370     if (!isBT) {
371         document.getElementById('specialty').value = '';
372         document.getElementById('carrera').innerHTML = '<option value="">Sin especificar</option>';
373     }
374
375     updateNocturnaVisibility(type, '');
376     generateCourseName();
377 }
378
379 function onGradeChange() {
380     const type = document.getElementById('education_type').value;
381     const grade = document.getElementById('grade_level').value;
382     updateNocturnaVisibility(type, grade);
383     generateCourseName();
384 }
385
386 function updateCarreras() {
387     const fig = document.getElementById('specialty').value;
388     const carreraSelect = document.getElementById('carrera');
389     carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
390
391     if (carreras[fig]) {
392         carreras[fig].forEach(function(c) {

```



```

392         const opt = document.createElement('option');
393         opt.value = c;
394         opt.textContent = c;
395         carreraSelect.appendChild(opt);
396     });
397 }
398 generateCourseName();
399 }
400
401 function generateCourseName() {
402     const grade = document.getElementById('grade_level').value;
403     const parallel = document.getElementById('parallel').value;
404     const shiftSelect = document.getElementById('shift_id');
405     const shiftOpt = shiftSelect.options[shiftSelect.selectedIndex];
406     const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';
407     const type = document.getElementById('education_type').value;
408     const specialty = document.getElementById('specialty').value;
409     const carrera = document.getElementById('carrera').value;
410
411     if (!grade || !parallel || !shiftName) {
412         document.getElementById('course_name').value = '';
413         return;
414     }
415
416     let name = grade + ' ' + parallel + ' ';
417
418     if (type === 'bt' && specialty) {
419         name += ' - ' + specialty;
420         if (carrera) name += ' (' + carrera + ')';
421     }
422
423     name += ' - ' + shiftName.charAt(0).toUpperCase() + shiftName.slice(1);
424     document.getElementById('course_name').value = name;
425 }
426 </script>
427
428 <!-- Crear Asignatura -->
429 <div class="card">
430     <h2>Crear Asignatura</h2>
431     <form method="POST" action="?action=create_subject">
432         <div class="form-group">
433             <label>Nombre</label>
434             <input type="text" name="name" placeholder="Ej: Matemáticas" required>
435         </div>
436         <div class="form-group">
437             <label>Código</label>
438             <input type="text" name="code" placeholder="Ej: MAT" required>
439         </div>
440         <button type="submit">Crear Asignatura</button>
441     </form>
442 </div>
443 </div>
444
445 <!-- Lista de Cursos -->
446 <div class="card">
447     <h2>Cursos Registrados</h2>
448     <table>
449         <thead>
450             <tr>
451                 <th>ID</th>
452                 <th>Nombre</th>
453                 <th>Nivel</th>
454                 <th>Paralelo</th>
455                 <th>Jornada</th>
456                 <th>Acciones</th>
457             </tr>
458         </thead>
459         <tbody>
460             <?php foreach($courses as $course): ?>
461                 <tr>
462                     <td>? $course['id'] ?></td>
463                     <td>? htmlspecialchars($course['name']) ?></td>
464                     <td>? htmlspecialchars($course['grade_level']) ?></td>
465                     <td>? htmlspecialchars($course['parallel']) ?></td>
466                     <td>? ucfirst($course['shift_name']) ?></td>
467                     <td style="white-space: nowrap;">
468                         <button onclick="location.href='?action=view_course_students&course_id=? $course...'
469                             style="padding: 5px 10px; font-size: 12px; background: #007bff;">
470                             ■ Estudiantes

```

```

471         </button>;
472         <button onclick="location.href='?action=edit_course&id=<?=$course['id'] ?>'"
473             style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #000;">
474             ■ Editar
475         </button>;
476         <form method="POST" action="?action=delete_course" style="display: inline;"
477             onsubmit="return confirmDeleteCourse(event, '<?=$course['id'] ?>'"
478             <input type="hidden" name="course_id" value="<?=$course['id'] ?>";
479         <button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;...
480             ■ Eliminar

481         </button>;
482         </form>;
483         </td>;
484         </tr>;
485         <?php endforeach; ?>
486     </tbody>;
487 </table>;
488 <button onclick="location.href='?action=enroll_students'" class="btn-primary" style="margin-top: 15px;"&g...
489     Matricular Estudiantes
490 </button>;
491 </div>;
492
493 <!-- Lista de Asignaturas -->;
494 <div class="card">;
495     <h2>Asignaturas Registradas</h2>;
496     <table>;
497         <thead>;
498             <tr>;
499                 <th>ID</th>;
500                 <th>Código</th>;
501                 <th>Nombre</th>;
502                 <th>Acciones</th>;
503             </tr>;
504         </thead>;
505         <tbody>;
506             <?php foreach($subjects as $subject): ?>
507                 <tr>;
508                     <td><?=$subject['id'] ?></td>;
509                     <td><?=$subject['code'] ?></td>;
510                     <td><?=$subject['name'] ?></td>;
511                     <td style="white-space: nowrap;">;
512                         <button onclick="location.href='?action=edit_subject&id=<?=$subject['id'] ?>'"...
513                             style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #000;">
514                             ■ Editar
515                         </button>;
516                         <form method="POST" action="?action=delete_subject" style="display: inline;"
517                             onsubmit="return confirmDeleteSubject(event, '<?=$subject['id'] ?>'"
518                             <input type="hidden" name="subject_id" value="<?=$subject['id'] ?>";
519                         <button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;...
520                             ■ Eliminar
521                         </button>;
522                         </form>;
523                     </td>;
524                 </tr>;
525                 <?php endforeach; ?>
526             </tbody>;
527         </table>;
528     </div>;
529 </div>;
530
531 </script>;
532 function confirmDeleteCourse(event, courseName) {
533     event.preventDefault();
534     const modal = document.createElement('div');
535     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
536     const modalContent = document.createElement('div');
537     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
538     modalContent.innerHTML = `
539         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Curso</h3>;
540         <p style="margin: 0 0 20px 0; color: #666;">¿Está seguro de eliminar el curso <strong>${courseNa...
541         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;"><strong>Nota:</strong> No se ...
542         <div style="display: flex; gap: 10px; justify-content: flex-end;">;
543             <button type="button" id="cancelCourseBtn" style="padding: 10px 20px; background: #6c757d; color: whi...
544             <button type="button" id="confirmCourseBtn" style="padding: 10px 20px; background: #dc3545; color: wh...
545         </div>;
546     modal.appendChild(modalContent);
547     document.body.appendChild(modal);
548     const form = event.target;
549     document.getElementById('confirmCourseBtn').onclick = function() { document.body.removeChild(modal); form.submit...

```

```

550     document.getElementById('cancelCourseBtn').onclick = function() { document.body.removeChild(modal); };
551     return false;
552 }
553
554 function confirmDeleteSubject(event, subjectName) {
555     event.preventDefault();
556     const modal = document.createElement('div');
557     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
558     const modalContent = document.createElement('div');
559     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
560     modalContent.innerHTML = `
561
562         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Asignatura</h3>
563         <p style="margin: 0 0 20px 0; color: #666;">¿Está seguro de eliminar la asignatura <strong>${sub...
564         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;"><strong>Nota:</strong> No se ...
565         <div style="display: flex; gap: 10px; justify-content: flex-end;">
566             <button type="button" id="cancelSubBtn" style="padding: 10px 20px; background: #6c757d; color: white;...
567             <button type="button" id="confirmSubBtn" style="padding: 10px 20px; background: #dc3545; color: white...
568         </div>`;
569     modal.appendChild(modalContent);
570     document.body.appendChild(modal);
571     const form = event.target;
572     document.getElementById('confirmSubBtn').onclick = function() { document.body.removeChild(modal); form.submit();...
573     document.getElementById('cancelSubBtn').onclick = function() { document.body.removeChild(modal); };
574     return false;
575 }
576
577 function confirmDelete(event, yearName) {
578     event.preventDefault();
579     const modal = document.createElement('div');
580     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
581     const modalContent = document.createElement('div');
582     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
583     modalContent.innerHTML = `
584         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Año Lectivo</h3>
585         <p style="margin: 0 0 20px 0; color: #666;">¿Está seguro de eliminar el año lectivo <strong>${ye...
586         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;"><strong>Nota:</strong> No se ...
587         <div style="display: flex; gap: 10px; justify-content: flex-end;">
588             <button type="button" id="cancelYearBtn" style="padding: 10px 20px; background: #6c757d; color: white...
589             <button type="button" id="confirmYearBtn" style="padding: 10px 20px; background: #dc3545; color: whit...
590         </div>`;
591     modal.appendChild(modalContent);
592     document.body.appendChild(modal);
593     const form = event.target;
594     document.getElementById('confirmYearBtn').onclick = function() { document.body.removeChild(modal); form.submit();...
595     document.getElementById('cancelYearBtn').onclick = function() { document.body.removeChild(modal); };
596     return false;
597 }
598 </script>
599 </body>
600 </html>

```

## 49. subject\_edit.php

Ruta: 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($subject['name']) ?>&lt;...
28                     </div>
29                     <div class="card-body">
30                         <?php if (!empty($errors)): ?>
31                             <div class="alert alert-danger">
32                                 <?php foreach ($errors as $error): ?>
33                                     <div><?= htmlspecialchars($error) ?>&lt;/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&lt;/small>
58                             </div>
59
60                             <div class="d-flex justify-content-between mt-4">
61                                 <a href="?action=academic" class="btn btn-secondary">← Cancelar&lt;/a>
62                                 <button type="submit" class="btn btn-success">✓ Guardar Cambios&lt;/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">&lt;/script>
72 </body>
73 </html>
```

## 50. index.php

Ruta: ArchImp/views/assignments/index.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Asignaciones Docentes - 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); 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: #28a745; color: white; border: none; border-radius: 4px; cursor: pointe...
17        button:hover { background: #218838; }
18        .btn-danger { background: #dc3545; padding: 5px 10px; font-size: 12px; }
19        .btn-danger:hover { background: #c82333; }
20        table { width: 100%; border-collapse: collapse; margin-top: 15px; }
21        th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
22        th { background: #f8f9fa; font-weight: bold; }
23        .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
24        .badge { padding: 4px 8px; border-radius: 3px; font-size: 11px; background: #ffc107; color: #333; }
25        h2 { margin-bottom: 20px; color: #333; }
26    </style>
27 </head>
28 <body>
29     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
30
31     <div class="container">
32         <?php if(isset($_GET['success'])): ?>
33             <div class="success">✓ Asignación creada correctamente</div>
34         <?php endif; ?>
35         <?php if(isset($_GET['error'])): ?>
36             <div class="error">✗ <?= htmlspecialchars($_GET['error']) ?></div>
37         <?php endif; ?>
38         <?php if(isset($_GET['removed'])): ?>
39             <div class="success">✓ Asignación eliminada</div>
40         <?php endif; ?>
41
42         <div class="card">
43             <h2>Asignar Docente a Materia</h2>
44             <form method="POST" action="?action=create_assignment">
45                 <div class="form-group">
46                     <label>Docente</label>
47                     <select name="teacher_id" required>
48                         <option value="">Seleccionar...</option>
49                         <?php foreach($teachers as $teacher): ?>
50                             <option value="<?= $teacher['id'] ?>">
51                                 <?= $teacher['last_name'] . ' ' . $teacher['first_name'] ?>
52                             </option>
53                         <?php endforeach; ?>
54                     </select>
55                 </div>
56
57                 <div class="form-group">
58                     <label>Curso</label>
59                     <select name="course_id" required>
60                         <option value="">Seleccionar...</option>
61                         <?php foreach($courses as $course): ?>
62                             <option value="<?= $course['id'] ?>">
63                                 <?= htmlspecialchars($course['name']) ?>
64                             </option>
65                         <?php endforeach; ?>
66                     </select>
67                 </div>
68
69                 <div class="form-group">
70                     <label>Asignatura</label>
71                     <select name="subject_id" required>
72                         <option value="">Seleccionar...</option>
73                         <?php foreach($subjects as $subject): ?>
74                             <option value="<?= $subject['id'] ?>">
75                                 <?= $subject['name'] ?></option>
76                         <?php endforeach; ?>
77                     </select>
78                 </div>
79
80                 <div class="form-group">
81                     <input type="submit" value="Asignar" />
82                 </div>
83             </form>
84         </div>
85     </div>
86 </body>
87 </html>
```

```

76         </select>;
77     </div>;
78
79     <button type="submit">Asignar</button>;
80 </form>;
81 </div>;
82
83 <div class="card">
84     <h2>Asignaciones Docente-Materia</h2>;
85
86     <!-- Filtros -->;
87     <form method="GET" action="" style="margin-bottom: 20px;">
88         <input type="hidden" name="action" value="assignments">;
89         <div style="display: grid; grid-template-columns: repeat(3, 1fr) auto; gap: 15px; align-items: flex-e...
90             <div class="form-group" style="margin-bottom: 0;">
91                 <label>Filtrar por Curso</label>;
92                 <select name="filter_course" onchange="this.form.submit()">
93                     <option value="">Todos los cursos</option>;
94                     <?php foreach($courses as $course): ?>
95                         <option value="<?=$course['id'] ?>" <?=(isset($_GET['filter_course'])) &...
96                             <?=$course['name'] ?>;
97                     </option>;
98                     <?php endforeach; ?>;
99                 </select>;
100             </div>;
101
102             <div class="form-group" style="margin-bottom: 0;">
103                 <label>Filtrar por Asignatura</label>;
104                 <select name="filter_subject" onchange="this.form.submit()">
105                     <option value="">Todas las asignaturas</option>;
106                     <?php foreach($subjects as $subject): ?>
107                         <option value="<?=$subject['id'] ?>" <?=(isset($_GET['filter_subject'])) &a...
108                             <?=$subject['name'] ?>;
109                     </option>;
110                     <?php endforeach; ?>;
111                 </select>;
112             </div>;
113
114             <div class="form-group" style="margin-bottom: 0;">
115                 <label>Filtrar por Docente</label>;
116                 <select name="filter_teacher" onchange="this.form.submit()">
117                     <option value="">Todos los docentes</option>;
118                     <?php foreach($teachers as $teacher): ?>
119                         <option value="<?=$teacher['id'] ?>" <?=(isset($_GET['filter_teacher'])) &a...
120                             <?=$teacher['last_name'] . ' ' . $teacher['first_name'] ?>;
121                     </option>;
122                     <?php endforeach; ?>;
123                 </select>;
124             </div>;
125
126             <?php if(isset($_GET['filter_course']) || isset($_GET['filter_subject']) || isset($_GET['filter_t...
127             <div>
128                 <a href="?action=assignments" style="padding: 10px 15px; background: #6c757d; color: white; t...
129                 Limpiar
130                 </a>;
131             </div>;
132             <?php endif; ?>;
133         </div>;
134     </form>;
135
136 </table>;
137 <thead>
138     <tr>
139         <th>Curso</th>;
140         <th>Asignatura</th>;
141         <th>Docente</th>;
142         <th>Acción</th>;
143     </tr>
144 </thead>
145 <tbody>
146     <?php
147     // Aplicar filtros
148     $filteredAssignments = $assignments;
149
150     if(isset($_GET['filter_course']) &amp;&amp; $_GET['filter_course'] != '') {
151         $filterCourseId = (int)$_GET['filter_course'];
152         $courseName = '';
153         foreach($courses as $c) {
154             if($c['id'] == $filterCourseId) {

```

```

155         $courseName = $c['name'];
156         break;
157     }
158 }
159 $filteredAssignments = array_filter($filteredAssignments, function($a) use ($courseName) {
160     return $a['course_name'] == $courseName;
161 });
162 }
163
164 if(isset($_GET['filter_subject']) && $_GET['filter_subject'] != '') {
165     $filterSubjectId = (int)$_GET['filter_subject'];
166     $subjectName = '';
167     foreach($subjects as $s) {
168         if($s['id'] == $filterSubjectId) {
169             $subjectName = $s['name'];
170             break;
171         }
172     }
173     $filteredAssignments = array_filter($filteredAssignments, function($a) use ($subjectName) {
174         return $a['subject_name'] == $subjectName;
175     });
176 }
177
178 if(isset($_GET['filter_teacher']) && $_GET['filter_teacher'] != '') {
179     $filterTeacherId = (int)$_GET['filter_teacher'];
180     $teacherName = '';
181     foreach($teachers as $t) {
182         if($t['id'] == $filterTeacherId) {
183             $teacherName = $t['last_name'] . ' ' . $t['first_name'];
184             break;
185         }
186     }
187     $filteredAssignments = array_filter($filteredAssignments, function($a) use ($teacherName) {
188         return $a['teacher_name'] == $teacherName;
189     });
190 }
191 ?>
192
193 <?php if(empty($filteredAssignments)): ?>
194     <tr>
195         <td colspan="4" style="text-align:center;">No hay asignaciones que coincidan con los f...
196     </tr>
197 <?php else: ?>
198     <?php foreach($filteredAssignments as $assignment): ?>
199         <tr>
200             <td><? $assignment['course_name'] ?></td>
201             <td><? $assignment['subject_name'] ?></td>
202             <td><? $assignment['teacher_name'] ?></td>
203             <td>
204                 <form method="POST" action="?action=remove_assignment" style="display: inline;">
205                     <input type="hidden" name="assignment_id" value="<? $assignment['id'] ?>.">...
206                     <button type="submit" class="btn-danger" onclick="confirmDeleteAssignment(event, ...
207                         Eliminar
208                     </button>
209                 </form>
210             </td>
211         </tr>
212     <?php endforeach; ?>
213 <?php endif; ?>
214 </tbody>
215 </table>
216 </div>
217 </div>
218
219 <script>
220 function confirmDeleteAssignment(event, teacherName, courseName, subjectName) {
221     event.preventDefault();
222
223     const modal = document.createElement('div');
224     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
225
226     const modalContent = document.createElement('div');
227     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
228
229     modalContent.innerHTML = `
230         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Asignación</h3>
231         <p style="margin: 0 0 20px 0; color: #666;">
232             ¿Está seguro de eliminar esta asignación?<br><br>
233             <strong>Docente:</strong> ${teacherName}<br>

```

```

234         <strong>Curso:</strong> ${courseName}<br>
235         <strong>Asignatura:</strong> ${subjectName}
236     </p>
237     <div style="display: flex; gap: 10px; justify-content: flex-end;">
238         <button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: whi...
239             Cancelar
240         </button>

241         <button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: wh...
242             Sí, Eliminar
243         </button>
244     </div>
245 `;
246
247     modal.appendChild(modalContent);
248     document.body.appendChild(modal);
249
250     const form = event.target.closest('form');
251
252     document.getElementById('confirmDeleteBtn').onclick = function() {
253         document.body.removeChild(modal);
254         form.submit();
255     };
256
257     document.getElementById('cancelDeleteBtn').onclick = function() {
258         document.body.removeChild(modal);
259     };
260
261     return false;
262 }
263 </script>
264 </body>
265 </html>

```



**Ruta:** 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: #c82323; }
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="container">
30         <?php if(isset($_GET['tutor_success'])): ?>
31             <div class="success">✓ Tutor asignado correctamente</div>
32         <?php endif; ?>
33         <?php if(isset($_GET['tutor_error'])): ?>
34             <div class="error">✗ <?= htmlspecialchars($_GET['tutor_error']) ?></div>
35         <?php endif; ?>
36         <?php if(isset($_GET['tutor_removed'])): ?>
37             <div class="success">✓ Tutor eliminado correctamente</div>
38         <?php endif; ?>
39
40         <div class="card">
41             <h2>Asignar Docente Tutor</h2>
42             <form method="POST" action="?action=set_tutor" id="tutorForm">
43                 <div class="form-group">
44                     <label>Curso</label>
45                     <select name="course_id" id="course_tutor" required>
46                         <option value="">Seleccionar...</option>
47                         <?php foreach($courses as $course): ?>
48                             <option value="<?=$course['id']"><?=$course['name']"> - <?=$course['shift_name']">
49                         </option>
50                     <?php endforeach; ?>
51                 </select>
52             </div>
53
54             <div class="form-group">
55                 <label>Docente Tutor</label>
56                 <select name="teacher_id" id="teacher_tutor" required>
57                     <option value="">Primero seleccione un curso...</option>
58                 </select>
59             </div>
60
61             <button type="submit">Asignar Tutor</button>
62         </form>
63
64         <hr style="margin: 30px 0; border: none; border-top: 1px solid #ddd;">
65     </div>
66
67     <div class="card">
68         <h2>Gestión de Tutores por Curso</h2>
69         <table>
70             <thead>
71                 <tr>
72                     <th>Curso</th>
73                     <th>Docente Tutor</th>
74                 </tr>
75             </thead>

```

```

6      <th>Acción</th>
7      </tr>
8      </thead>
9      <tbody>
10         <?php foreach($courses as $course): ?>
11
12             <?php
13                 // Buscar si el curso tiene tutor
14                 $courseTutor = null;
15                 foreach($assignments as $assignment) {
16                     if ($assignment['course_name'] == $course['name'] && $assignment['is_tutor']) {
17                         $courseTutor = $assignment;
18                         break;
19                     }
20                 }
21             ?>
22             <tr>
23                 <td><?php if($courseTutor): ?>
24                     <input type="hidden" name="course_id" value="<?php echo $course['id']; ?>" />
25                     <button type="submit" class="btn-danger" style="padding: 5px 10px; font-size: 1em;">Quitar</button>
26                 <?php else: ?>
27                     -
28                 <?php endif; ?>
29             </td>
30         </tr>
31     </tbody>
32 </table>
33 </div>
34 </div>
35 </div>
36 </script>
37
38 let currentTutor = null;
39
40 document.getElementById('course_tutor').addEventListener('change', function() {
41     const courseId = this.value;
42     const teacherSelect = document.getElementById('teacher_tutor');
43     currentTutor = null;
44
45     teacherSelect.innerHTML = '<option value="">Cargando...</option>';
46
47     if (!courseId) {
48         teacherSelect.innerHTML = '<option value="">Primero selecciona un curso...</option>';
49         return;
50     }
51
52     fetch('?action=check_course_tutor&course_id=' + courseId)
53         .then(response => response.json())
54         .then(tutor => {
55             if (tutor && tutor.tutor_name) {
56                 currentTutor = tutor.tutor_name;
57             }
58         });
59
60     fetch('?action=get_course_teachers&course_id=' + courseId)
61         .then(response => response.json())
62         .then(data => {
63             if (data.length === 0) {
64                 teacherSelect.innerHTML = '<option value="">No hay docentes asignados a este curso</option>';
65             } else {
66                 teacherSelect.innerHTML = '<option value="">Seleccionar...</option>';
67                 data.forEach(teacher => {
68                     const option = document.createElement('option');
69                     option.value = teacher.teacher_id;
70                     option.textContent = teacher.teacher_name;
71                     teacherSelect.appendChild(option);
72                 });
73             }
74         });
75
76 });

```

```

155         });
156     }
157 }
158 .catch(error => {
159     teacherSelect.innerHTML = '<option value="">Error al cargar docentes</option>';
160     console.error('Error:', error);
161 });
162 });
163
164 document.getElementById('tutorForm').addEventListener('submit', function(e) {
165     if (currentTutor) {
166         e.preventDefault();
167
168         const modal = document.createElement('div');
169         modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0...
170
171         const modalContent = document.createElement('div');
172         modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh...
173
174         modalContent.innerHTML = `
175             <h3 style="margin: 0 0 15px 0; color: #333;">■ Confirmar Cambio de Tutor</h3>
176             <p style="margin: 0 0 20px 0; color: #666;">
177                 El curso ya tiene como tutor a: <strong>${currentTutor}</strong>.<br>
178                 ¿Desea reemplazarlo por el nuevo docente?
179             </p>
180             <div style="display: flex; gap: 10px; justify-content: flex-end;">
181                 <button type="button" id="cancelBtn" style="padding: 10px 20px; background: #6c757d; color: white...
182                     Cancelar
183                 </button>
184                 <button type="button" id="confirmBtn" style="padding: 10px 20px; background: #007bff; color: whit...
185                     Sí, Reemplazar
186                 </button>
187             </div>
188         `;
189
190         modal.appendChild(modalContent);
191         document.body.appendChild(modal);
192
193         const form = this;
194
195         document.getElementById('confirmBtn').onclick = function() {
196             document.body.removeChild(modal);
197             currentTutor = null;
198             form.submit();
199         };
200
201         document.getElementById('cancelBtn').onclick = function() {
202             document.body.removeChild(modal);
203         };
204     }
205 });
206
207 function confirmRemoveTutor(event, tutorName, courseName) {
208     event.preventDefault();
209
210     const modal = document.createElement('div');
211     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
212
213     const modalContent = document.createElement('div');
214     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
215
216     modalContent.innerHTML = `
217         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Quitar Tutor</h3>
218         <p style="margin: 0 0 20px 0; color: #666;">
219             ¿Está seguro de quitar a <strong>${tutorName}</strong> como tutor del curso <strong>${...
220         </p>
221         <div style="display: flex; gap: 10px; justify-content: flex-end;">
222             <button type="button" id="cancelRemoveBtn" style="padding: 10px 20px; background: #6c757d; color: whi...
223                 Cancelar
224             </button>
225             <button type="button" id="confirmRemoveBtn" style="padding: 10px 20px; background: #dc3545; color: wh...
226                 Sí, Quitar
227             </button>
228         </div>
229     `;
230
231     modal.appendChild(modalContent);
232     document.body.appendChild(modal);
233

```

```
234     const form = event.target;
235
236     document.getElementById('confirmRemoveBtn').onclick = function() {
237         document.body.removeChild(modal);
238         form.submit();
239     };
240
241     document.getElementById('cancelRemoveBtn').onclick = function() {
242         document.body.removeChild(modal);
243     };
244
245     return false;
246 }
247 </script>
248 </body>
249 </html>
```

## 52. view\_course.php

Ruta: 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-bot...
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-bott...
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> <? $course['grade_level'] ?> | <strong>Paralel...
37                 <p><strong>Jornada:</strong> <? ucfirst($course['shift_name']) ?> | <strong>...
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>
```

## 53. calendar.php

Ruta: ArchImp/views/attendance/calendar.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Calendario de 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-bot...
15        .calendar { display: grid; grid-template-columns: repeat(7, 1fr); gap: 10px; margin-top: 20px; }
16        .calendar-header { text-align: center; font-weight: bold; padding: 10px; background: #f8f9fa; }
17        .calendar-day { border: 1px solid #ddd; padding: 10px; min-height: 80px; border-radius: 4px; position: relative;...
18        .calendar-day.today { background: #fff3cd; border-color: #ffc107; }
19        .calendar-day.has-attendance { background: #d4edda; }
20        .calendar-day.weekend { background: #f8f9fa; color: #999; }
21        .day-number { font-weight: bold; margin-bottom: 5px; }
22        .attendance-summary { font-size: 11px; color: #666; }
23        .month-nav { display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px; }
24        .btn { padding: 8px 16px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointer; ...
25        .btn:hover { background: #0056b3; }
26        .form-group { margin-bottom: 15px; }
27        select { padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
28    </style>
29 </head>
30 <body>
31     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
32
33     <div class="container">
34         <div class="card">
35             <div class="form-group">
36                 <label>Seleccionar Curso</label>
37                 <select id="course-select" onchange="loadCalendar()">
38                     <option value="">Seleccionar curso...</option>
39                     <?php foreach($courses as $course): ?>
40                         <option value="<?=$course['id'] ?>" <?=$course['name'] ?> & & $course['shift_name'] ?>
41                     </option>
42                 </select>
43             </div>
44
45             <?php if(isset($_GET['course_id'])): ?>
46                 <div class="month-nav">
47                     <a href="?action=attendance_calendar&course_id=<?=$_GET['course_id'] ?>&month=<?=$_GET['month'] ?>">
48                         <?=$_GET['month'] ?>
49                     </a>
50                     <a href="?action=attendance_calendar&course_id=<?=$_GET['course_id'] ?>&month=<?=$_GET['month'] ?>">
51                         <?=$_GET['month'] ?>
52                     </a>
53                 </div>
54
55                 <div class="calendar">
56                     <div class="calendar-header">
57                         <?php foreach(['Dom', 'Lun', 'Mar', 'Mié', 'Jue', 'Vie', 'Sáb'] as $day): ?>
58                             <div class="calendar-day">
59                                 <div class="day-number"><?=$day ?></div>
60                                 <div class="attendance-summary">
61                                     <?=$day['attendance']['presente'] ?>
62                                     <?=$day['attendance']['ausente'] ?>
63                                 </div>
64                             </div>
65                         </div>
66                     <div class="calendar-day">
67                         <div class="day-number"><?=$day ?></div>
68                         <div class="attendance-summary">
69                             <?=$day['attendance']['presente'] ?>
70                             <?=$day['attendance']['ausente'] ?>
71                         </div>
72                     </div>
73                 </div>
74             </div>
75         </div>
76     </div>
```

```
76
77     <script>
78         function loadCalendar() {
79             const courseId = document.getElementById('course-select').value;
80             if (courseId) {
81
82                 window.location.href = '?action=attendance_calendar&course_id=' + courseId;
83             }
84         }
85     </script>
86 </body>
</html>
```



## 54. course\_view.php

Ruta: 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>Asistencia del Curso - EcuAsist</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9     <style>
10         body { padding-top: 70px; background: #f4f4f4; }
11         .badge-presente { background-color: #28a745; }
12         .badge-ausente { background-color: #dc3545; }
13         .badge-tardanza { background-color: #ffc107; color: #000; }
14         .badge-justificado { background-color: #17a2b8; }
15     </style>
16 </head>
17 <body>
18     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
19
20     <div class="container mt-4">
21
22         <?php if(empty($courses)): ?>
23             <div class="alert alert-warning">
24                 <h5>■ Sin Cursos Asignados</h5>
25                 <p class="mb-0">
26                     <?php if(Security::hasRole('docente')): ?>
27                         No tiene cursos asignados como tutor ni asignaturas asignadas.
28                         Contacte con la autoridad para que le asignen cursos.
29                     <?php else: ?>
30                         No hay cursos disponibles en el sistema.
31                     <?php endif; ?>
32                 </p>
33             </div>
34         <?php else: ?>
35
36             <div class="card mb-4">
37                 <div class="card-header bg-primary text-white">
38                     <h5 class="mb-0">■ Asistencia General del Curso</h5>
39                     <?php if(Security::hasRole('docente')): ?>
40                         <small>Mostrando solo cursos donde es tutor o tiene asignaturas</small>
41                     <?php endif; ?>
42                 </div>
43                 <div class="card-body">
44                     <form method="GET" id="filterForm">
45                         <input type="hidden" name="action" value="view_course_attendance">
46                         <div class="row">
47                             <div class="col-md-4 mb-3">
48                                 <label class="form-label">Curso *</label>
49                                 <select name="course_id" class="form-select" required>
50                                     <option value="">Seleccionar curso...</option>
51                                     <?php foreach($courses as $c):
52                                         $isTutor = isset($c['is_tutor']) && $c['is_tutor'];
53                                         $label = html_entity_decode($c['name'], ENT_QUOTES, 'UTF-8');
54                                         if ($isTutor) $label .= ' ■ (Mi curso / Tutor)';
55                                     <?>
56                                     <option value="<?=$c['id']">
57                                         <?=$isTutor && $courseId == $c['id'] ? 'selected' : '' ?>...
58                                         <?=$label ?>
59                                     </option>
60                                 <?php endforeach; ?>
61                             </select>
62                         </div>
63
64                         <div class="col-md-3 mb-3">
65                             <label class="form-label">Fecha Inicio</label>
66                             <input type="date" name="start_date" class="form-control"
67                                 value="<?=$startDate ?? date('Y-m-01')">
68                         </div>
69
70                         <div class="col-md-3 mb-3">
71                             <label class="form-label">Fecha Fin</label>
72                             <input type="date" name="end_date" class="form-control"
73                                 value="<?=$endDate ?? date('Y-m-d')">
74                         </div>
75
```

```

76         <div class="col-md-2 mb-3">
77             <label class="form-label">&nbsp;</label>
78             <button type="submit" class="btn btn-primary w-100">
79                 ■ Consultar
80             </button>
81         </div>
82     </div>
83 </form>
84 </div>
85 </div>
86
87 <?php if(isset($courseId) && !empty($attendanceData)): ?>
88
89 <!-- Resumen estadístico -->
90 <div class="row mb-4">
91     <?php
92     $totalPresente = array_sum(array_column($attendanceData, 'presente'));
93     $totalAusente = array_sum(array_column($attendanceData, 'ausente'));
94     $totalTardanza = array_sum(array_column($attendanceData, 'tardanza'));
95     $totalJustificado = array_sum(array_column($attendanceData, 'justificado'));
96     $totalRegistros = array_sum(array_column($attendanceData, 'total_registros'));
97     ?>
98
99     <div class="col-md-3">
100         <div class="card text-center border-success">
101             <div class="card-body">
102                 <h3 class="text-success">&nbsp;= $totalPresente ?&nbsp;</h3>
103                 <p class="mb-0">Presentes</p>
104             </div>
105         </div>
106     </div>
107
108     <div class="col-md-3">
109         <div class="card text-center border-danger">
110             <div class="card-body">
111                 <h3 class="text-danger">&nbsp;= $totalAusente ?&nbsp;</h3>
112                 <p class="mb-0">Ausentes</p>
113             </div>
114         </div>
115     </div>
116
117     <div class="col-md-3">
118         <div class="card text-center border-warning">
119             <div class="card-body">
120                 <h3 class="text-warning">&nbsp;= $totalTardanza ?&nbsp;</h3>
121                 <p class="mb-0">Tardanzas</p>
122             </div>
123         </div>
124     </div>
125
126     <div class="col-md-3">
127         <div class="card text-center border-info">
128             <div class="card-body">
129                 <h3 class="text-info">&nbsp;= $totalJustificado ?&nbsp;</h3>
130                 <p class="mb-0">Justificados</p>
131             </div>
132         </div>
133     </div>
134 </div>
135
136 <!-- Tabla de estudiantes -->
137 <div class="card">
138     <div class="card-header bg-light d-flex justify-content-between align-items-center">
139         <h5 class="mb-0">Estudiantes del Curso</h5>
140         <span class="badge bg-primary">&nbsp;= count($attendanceData) ?&nbsp; estudiantes</span>
141     </div>
142     <div class="card-body">
143         <div class="alert alert-info">
144             <strong>■ Período:</strong>
145             &nbsp;= date('d/m/Y', strtotime($startDate)) ?&nbsp; al &nbsp;= date('d/m/Y', strtotime($endDate)) ?&nbsp;
146         </div>
147
148         <div class="table-responsive">
149             <table class="table table-hover table-bordered">
150                 <thead class="table-light">
151                     <tr>
152                         <th>&nbsp;</th>
153                         <th>Estudiante</th>
154                         <th class="text-center">Total Registros</th>

```

```

155         <th class="text-center">Presentes</th>
156         <th class="text-center">Ausentes</th>
157         <th class="text-center">Tardanzas</th>
158         <th class="text-center">Justificados</th>
159         <th class="text-center">% Asistencia</th>
160     </tr>
161 </thead>
162 <tbody>
163     <?php
164         $counter = 1;
165         foreach($attendanceData as $data):
166             $percentage = $data['total_registros'] > 0
167                 ? round(($data['presente'] / $data['total_registros']) * 100, 1)
168                 : 0;
169
170             $percentageClass = '';
171             if ($percentage >= 90) $percentageClass = 'text-success fw-bold';
172             elseif ($percentage >= 75) $percentageClass = 'text-warning fw-bold';
173             else $percentageClass = 'text-danger fw-bold';
174         <?>
175         <tr>
176             <td><?php $counter++ <?></td>
177             <td><strong><?php $data['student_name'] <?></strong></td>
178             <td class="text-center"><?php $data['total_registros'] <?></td>
179             <td class="text-center">
180                 <span class="badge badge-presente"><?php $data['presente'] <?></span>
181             </td>
182             <td class="text-center">
183                 <span class="badge badge-ausente"><?php $data['ausente'] <?></span>
184             </td>
185             <td class="text-center">
186                 <span class="badge badge-tardanza"><?php $data['tardanza'] <?></span>
187             </td>
188             <td class="text-center">
189                 <span class="badge badge-justificado"><?php $data['justificado'] <?></span>
190             </td>
191             <td class="text-center <?php $percentageClass <?>>>
192                 <?php $percentage <?>%
193             </td>
194         </tr>
195     <?php endforeach; <?>
196 </tbody>
197 </table>
198 </div>
199
200 <div class="mt-3">
201     <small class="text-muted">
202         <strong>Nota:</strong> El porcentaje de asistencia se calcula sobre el total de regi...
203     </small>
204 </div>
205 </div>
206 </div>
207
208 <?php elseif(isset($courseId) && empty($attendanceData)): <?>
209 <div class="alert alert-warning text-center">
210     ■ No hay registros de asistencia para el curso seleccionado en el período indicado.
211 </div>
212
213 <?php else: <?>
214 <div class="alert alert-info text-center">
215     ■ Seleccione un curso y período para ver la asistencia.
216 </div>
217 <?php endif; <?>
218
219 <?php endif; // Fin de validación de cursos <?>
220 </div>
221
222 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
223 </body>
224 </html>

```

## 55. my\_attendance.php

Ruta: 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      <title>Mi 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); }
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(200px, 1fr)); gap: 20px; margin-bottom: 3...
24         .stat-card { color: white; padding: 20px; border-radius: 8px; text-align: center; }
25         .stat-number { font-size: 36px; font-weight: bold; margin-bottom: 5px; }
26         .stat-label { font-size: 14px; opacity: 0.9; }
27         .btn-justify {
28             color: #007bff;
29             text-decoration: none;
30             font-size: 12px;
31             font-weight: bold;
32         }
33         .btn-justify:hover {
34             text-decoration: underline;
35         }
36     </style>
37 </head>
38 <body>
39     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
40
41     <div class="container">
42         <?php
43         $total = count($attendances);
44         $presente = count(array_filter($attendances, fn($a) => $a['status'] === 'presente'));
45         $ausente = count(array_filter($attendances, fn($a) => $a['status'] === 'ausente'));
46         $tardanza = count(array_filter($attendances, fn($a) => $a['status'] === 'tardanza'));
47         $asistencia = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
48         ?>
49
50         <div class="stats">
51             <div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
52                 <div class="stat-number"><?=$total ?></div>
53                 <div class="stat-label">Total Registros</div>
54             </div>
55             <div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
56                 <div class="stat-number"><?=$asistencia ?>%</div>
57                 <div class="stat-label">Asistencia</div>
58             </div>
59             <div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
60                 <div class="stat-number"><?=$ausente ?></div>
61                 <div class="stat-label">Ausencias</div>
62             </div>
63             <div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
64                 <div class="stat-number"><?=$tardanza ?></div>
65                 <div class="stat-label">Tardanzas</div>
66             </div>
67         </div>
68
69         <div class="card">
70             <h2>Historial de Asistencia</h2>
71             <table>
72                 <thead>
73                     <tr>
74                         <th>Fecha</th>
75                         <th>Curso</th>
```

```

76         <th>Asignatura</th>;
77         <th>Jornada</th>;
78         <th>Hora</th>;
79         <th>Estado</th>;
80         <th>Observación</th>;

81         <th>Acciones</th>;
82     </tr>;
83 </thead>;
84 <tbody>;
85     <?php foreach($attendances as $att): ?>
86         <tr>;
87             <td><?php echo date('d/m/Y', strtotime($att['date'])) ?></td>;
88             <td><?php echo $att['course_name'] ?></td>;
89             <td><?php echo $att['subject_name'] ?></td>;
90             <td><?php echo ucfirst($att['shift_name']) ?></td>;
91             <td><?php echo $att['hour_period'] ?></td>;
92             <td>;
93                 <span class="badge badge-<?php echo $att['status'] ?>">
94                     <?php echo ucfirst($att['status']) ?>
95                 </span>;
96             </td>;
97             <td><?php echo $att['observation'] ?>: '-' ?></td>;
98             <td>;
99                 <?php if($att['status'] === 'ausente'): ?>
100                     <a href="?action=submit_justification&attendance_id=<?php echo $att['id'] ?>"
101                         class="btn-justify">
102                         Justificar
103                     </a>;
104                 <?php else: ?>
105                     -
106                 <?php endif; ?>
107             </td>;
108         </tr>;
109     <?php endforeach; ?>

110
111     <?php if(empty($attendances)): ?>
112         <tr>;
113             <td colspan="8" style="text-align:center;">No hay registros de asistencia.</td>;
114         </tr>;
115     <?php endif; ?>
116 </tbody>;
117 </table>;
118 </div>;
119 </div>;
120 </body>;
121 </html>;
122

```

## 56. register.php

Ruta: ArchImp/views/attendance/register.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Registrar Asistencia - 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: 20px; }
13        label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
14        input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
15        button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointe...
16        button:hover { background: #0056b3; }
17        table { width: 100%; border-collapse: collapse; margin-top: 20px; }
18        th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
19        th { background: #f8f9fa; }
20        .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
21        .error { background: #f8d7da; color: #721c24; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
22        #student-list { display: none; }
23        select.status-select { width: auto; padding: 5px; }
24        .warning { background: #fff3cd; color: #856404; padding: 10px; border-radius: 4px; margin-bottom: 15px; font-siz...
25        .class-card { border: 1px solid #ddd; padding: 15px; border-radius: 4px; margin-bottom: 10px; cursor: pointer; t...
26        .class-card:hover { background: #f8f9fa; border-color: #007bff; }
27        .class-card.selected { background: #e7f3ff; border-color: #007bff; }
28        .class-info { display: flex; justify-content: space-between; align-items: center; }
29        .class-details h3 { color: #007bff; margin-bottom: 5px; }
30        .class-details p { color: #666; font-size: 14px; }
31    </style>
32 </head>
33 <body>
34     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
35
36     <div class="container">
37         <?php if(isset($_GET['success'])): ?>
38             <div class="success">✓ Asistencia registrada correctamente</div>
39         <?php endif; ?>
40
41         <?php if(isset($_GET['error'])): ?>
42             <?php if($_GET['error'] == 'future'): ?>
43                 <div class="error">✗ No se puede registrar asistencia de fechas futuras</div>
44             <?php elseif($_GET['error'] == 'toolate'): ?>
45                 <div class="error">✗ Solo se puede registrar asistencia de hasta 48 horas atrás</div>
46             <?php endif; ?>
47         <?php endif; ?>
48
49         <div class="card">
50             <h2>Mis Clases de Hoy</h2>
51             <div class="warning">
52                 ■ <strong>Importante:</strong> Puede registrar asistencia de hoy o corregir registros anteri...
53             </div>
54
55             <?php if(empty($todayClasses)): ?>
56                 <p style="color: #666; text-align: center; padding: 30px;">
57                     No tienes clases programadas para hoy según tu horario.
58                 </p>
59             <?php else: ?>
60                 <div id="classes-list">
61                     <?php foreach($todayClasses as $class): ?>
62                         <?php
63                             $jsonCourse = json_encode($class['course_name'], JSON_HEX_TAG | JSON_HEX_APOS);
64                             $jsonSubject = json_encode($class['subject_name'], JSON_HEX_TAG | JSON_HEX_APOS);
65                         <?>
66                         <div class="class-card" onclick='selectClass(event, <?=$class["id"] ?>, <?=$jsonCourse...
67                             <div class="class-info">
68                                 <div class="class-details">
69                                     <h3><?=$class['period_number'] ?><?ra hora - <?=$class['subject_name'] ?&...
70                                     <p><?Curso: <?=$class['course_name'] ?></p>
71                                 </div>
72                             </div>
73                             <button type="button" style="padding: 8px 15px; font-size: 14px;">
74                                 Tomar Asistencia
75                             </button>
```

```

76         <div>
77             </div>
78         </div>
79         <?php endforeach; ?>
80     </div>
81
82     <?php endif; ?>
83 </div>
84 <div class="card" id="date-selector" style="display: none;">
85     <h2>Seleccionar Fecha</h2>
86     <form id="config-form">
87         <input type="hidden" id="schedule_id" name="schedule_id">
88         <div class="form-group">
89             <label>Clase Seleccionada</label>
90             <input type="text" id="selected_class_info" readonly style="background: #f0f0f0;">
91         </div>
92         <div class="form-group">
93             <label>Fecha</label>
94             <input type="date" id="date" name="date" value="<?php = date('Y-m-d') ?>"
95                 max="<?php = date('Y-m-d') ?>"
96                 min="<?php = $minDate ?>"
97                 required>
98             <small style="color: #666; display: block; margin-top: 5px;">
99                 ■ La asistencia de hoy podrá editarse hasta el <?php = date('d/m/Y', strtotime($maxEditDate)) ?>...
100             </small>
101         </div>
102         <button type="button" onclick="loadStudents()">Cargar Estudiantes</button>
103     </form>
104 </div>
105
106 <form method="POST" id="student-list">
107     <input type="hidden" name="schedule_id" id="schedule_id_hidden">
108     <input type="hidden" name="date" id="date_hidden">
109
110     <div class="card">
111         <h2>Lista de Estudiantes</h2>
112         <table id="students-table">
113             <thead>
114                 <tr>
115                     <th><?php = $minDate ?></th>
116                     <th>Apellidos y Nombres</th>
117                     <th>Cédula</th>
118                     <th>Estado</th>
119                     <th>Observación</th>
120                 </tr>
121             </thead>
122             <tbody><tbody>
123         </table>
124         <button type="submit" style="margin-top: 20px;">Guardar Asistencia</button>
125     </div>
126 </form>
127 </div>
128
129 <script>
130     const minDateAllowed = '<?php = $minDate ?>';
131     let selectedScheduleId = null;
132
133     function selectClass(event, scheduleId, courseName, subjectName, periodNumber) {
134         selectedScheduleId = scheduleId;
135
136         // Remover selección anterior
137         document.querySelectorAll('.class-card').forEach(card => {
138             card.classList.remove('selected');
139         });
140
141         // Seleccionar actual
142         event.currentTarget.classList.add('selected');
143
144         // Mostrar selector de fecha
145         document.getElementById('date-selector').style.display = 'block';
146         document.getElementById('schedule_id').value = scheduleId;
147         document.getElementById('selected_class_info').value = periodNumber + 'ra hora - ' + subjectName + ' (' + co...
148
149         // Ocultar lista de estudiantes
150         document.getElementById('student-list').style.display = 'none';
151     }
152
153     document.getElementById('date').addEventListener('change', function() {
154         const selectedDate = this.value;

```

```

155     const today = new Date().toISOString().split('T')[0];
156
157     if (selectedDate > today) {
158         alert('No se puede seleccionar una fecha futura');
159         this.value = today;
160         return;
161     }
162
163     if (selectedDate < minDateAllowed) {
164         alert('Solo puede registrar asistencia de hasta 48 horas hábiles atrás');
165         this.value = today;
166     }
167     });
168
169     function loadStudents() {
170         const scheduleId = document.getElementById('schedule_id').value;
171         const date = document.getElementById('date').value;
172
173         if (!scheduleId || !date) {
174             alert('Complete todos los campos');
175             return;
176         }
177
178         document.getElementById('schedule_id_hidden').value = scheduleId;
179         document.getElementById('date_hidden').value = date;
180
181         let courseId = null;
182
183         // Obtener course_id del horario
184         fetch('?action=get_schedule_info&schedule_id=' + scheduleId)
185             .then(response => response.json())
186             .then(scheduleData => {
187                 courseId = scheduleData.course_id;
188                 const formData = new FormData();
189                 formData.append('course_id', courseId);
190                 return fetch('?action=get_students', {
191                     method: 'POST',
192                     body: formData
193                 });
194             })
195             .then(response => response.json())
196             .then(students => {
197                 if (students.length === 0) {
198                     alert('No hay estudiantes matriculados en este curso');
199                     return;
200                 }
201
202                 // Consultar asistencia ya registrada para esta clase y fecha
203                 return fetch('?action=get_existing_attendance&schedule_id=' + scheduleId + '&date=' + date)
204                     .then(response => response.json())
205                     .then(existing => {
206                         // Convertir a mapa: student_id => {status, observation}
207                         const existingMap = {};
208                         existing.forEach(a => {
209                             existingMap[a.student_id] = { status: a.status, observation: a.observation || '' };
210                         });
211
212                         const tbody = document.querySelector('#students-table tbody');
213                         tbody.innerHTML = '';
214
215                         const statuses = ['presente', 'ausente', 'tardanza', 'justificado'];
216
217                         students.forEach((student, index) => {
218                             const current = existingMap[student.id] || { status: 'presente', observation: '' };
219                             const statusOptions = statuses.map(s =>
220                                 `<option value="${s}" ${current.status === s ? 'selected' : ''}>${s.charAt(0)...
221                                 ).join('');
222
223                             const row = `
224                                 <tr>
225                                     <td>${index + 1}</td>
226                                     <td>${student.last_name} ${student.first_name}</td>
227                                     <td>${student.dni || '-'}</td>
228                                     <td>
229                                         <select name="status_${student.id}" class="status-select">
230                                             ${statusOptions}
231                                         </select>
232                                     </td>
233                                 </tr>

```



```

234             <input type="text" name="obs_{$student.id}"
235                 value="{$current.observations}"
236                 placeholder="Opcional" style="width: 100%;">
237             </td>
238         </tr>
239     `;
240     tbody.innerHTML += row;

241     });
242
243     document.getElementById('student-list').style.display = 'block';
244     document.getElementById('student-list').scrollIntoView({ behavior: 'smooth' });
245     });
246 }
247 .catch(error => {
248     console.error('Error:', error);
249     // Diagnóstico
250     fetch('?action=get_existing_attendance&schedule_id=' + scheduleId + '&date=' + date)
251         .then(r => r.text())
252         .then(t => {
253             console.log('Respuesta get_existing_attendance:', t);
254         });
255     });
256 }
257 </script>
258 </body>
259 </html>

```

## 57. view.php

Ruta: ArchImp/views/attendance/view.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Ver Asistencias - 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-bot...
15        .form-group { margin-bottom: 15px; }
16        label { display: block; margin-bottom: 5px; font-weight: bold; }
17        input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
18        button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointe...
19        button:hover { background: #0056b3; }
20        table { width: 100%; border-collapse: collapse; margin-top: 20px; }
21        th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
22        th { background: #f8f9fa; }
23        .badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
24        .badge-presente { background: #28a745; }
25        .badge-ausente { background: #dc3545; }
26        .badge-tardanza { background: #ffc107; color: #333; }
27        .badge-justificado { background: #17a2b8; }
28    </style>
29 </head>
30 <body>
31     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
32
33     <div class="container">
34         <div class="card">
35             <h2>Consultar</h2>
36             <form method="POST">
37                 <div class="form-group">
38                     <label>Curso</label>
39                     <select name="course_id" required>
40                         <option value="">Seleccionar...</option>
41                         <?php foreach($courses as $course): ?>
42                             <option value="<?php echo $course['id']; ?>"><?php echo $course['name']; ?> - <?php echo $cour...
43                         <?php endforeach; ?>
44                     </select>
45                 </div>
46                 <div class="form-group">
47                     <label>Fecha</label>
48                     <input type="date" name="date" value="<?php echo date('Y-m-d'); ?>" required>
49                 </div>
50                 <button type="submit">Buscar</button>
51             </form>
52         </div>
53
54         <?php if(!empty($attendances)): ?>
55             <div class="card">
56                 <h2>Resultados</h2>
57                 <table>
58                     <thead>
59                         <tr>
60                             <th>Estudiante</th>
61                             <th>Asignatura</th>
62                             <th>Docente</th>
63                             <th>Hora</th>
64                             <th>Estado</th>
65                             <th>Observación</th>
66                         </tr>
67                     </thead>
68                     <tbody>
69                         <?php foreach($attendances as $att): ?>
70                             <tr>
71                                 <td><?php echo $att['last_name']; ?> <?php echo $att['first_name']; ?></td>
72                                 <td><?php echo $att['subject_name']; ?></td>
73                                 <td><?php echo $att['teacher_name']; ?></td>
74                                 <td><?php echo $att['hour_period']; ?></td>
75                                 <td></td>
```

```
76         <span class="badge badge-<?=$att['status'] ?>">
77             <?=$ucfirst($att['status']) ?>
78         </span>
79     </td>
80     <td><?=$att['observation'] ?: '-' ?></td>
81 </tr>
82 <?php endforeach; ?>
83 </tbody>
84 </table>
85 </div>
86 <?php endif; ?>
87 </div>
88 </body>
89 </html>
```

## 58. forgot.php

Ruta: 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>
```

## 59. login.php

Ruta: ArchImp/views/auth/login.php

```
1  <?php
2  if(isset($_SESSION['user_id'])) {
3      header('Location: ?action=dashboard');
4      exit;
5  }
6  ?>
7  <!DOCTYPE html>
8  <html lang="es">
9  <head>
10     <meta charset="UTF-8">
11     <meta name="viewport" content="width=device-width, initial-scale=1.0">
12     <title>Iniciar Sesión - EcuAsist 2026</title>
13     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
14     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.4.0/css/all.min.css">
15     <style>
16         body {
17             background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
18             min-height: 100vh;
19             display: flex;
20             align-items: center;
21             justify-content: center;
22         }
23         .login-card {
24             background: white;
25             border-radius: 15px;
26             box-shadow: 0 10px 40px rgba(0,0,0,0.2);
27             max-width: 420px;
28             width: 100%;
29         }
30     </style>
31 </head>
32 <body>
33     <div class="login-card p-5">
34         <div class="text-center mb-4">
35             <i class="fas fa-school fa-4x text-primary mb-3"></i>
36             <h2>EcuAsist 2026</h2>
37             <p class="text-muted">Sistema de Asistencia Escolar</p>
38         </div>
39
40         <?php if(isset($error)): ?>
41         <div class="alert alert-danger alert-dismissible fade show" role="alert">
42             <i class="fas fa-exclamation-circle"></i> <?= htmlspecialchars($error) ?>
43             <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
44         </div>
45         <?php endif; ?>
46
47         <?php if(isset($_GET['registered'])): ?>
48         <div class="alert alert-success alert-dismissible fade show" role="alert">
49             <i class="fas fa-check-circle"></i> Registro exitoso. Ahora puede iniciar sesión.
50             <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
51         </div>
52         <?php endif; ?>
53
54         <?php if(isset($_GET['reset'])): ?>
55         <div class="alert alert-success alert-dismissible fade show" role="alert">
56             <i class="fas fa-check-circle"></i> Contraseña restablecida correctamente.
57             <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
58         </div>
59         <?php endif; ?>
60
61         <form method="POST">
62             <?php Security::generateToken(); ?>
63             <input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">
64
65             <div class="mb-3">
66                 <label for="email" class="form-label">
67                     <i class="fas fa-user"></i> Usuario o Correo Electrónico
68                 </label>
69                 <input type="text" class="form-control" id="email" name="email" required autofocus placeholder="Ingre...
70             </div>
71
72             <div class="mb-3">
73                 <label for="password" class="form-label">
74                     <i class="fas fa-lock"></i> Contraseña
75                 </label>
```

```
76         <input type="password" class="form-control" id="password" name="password" required placeholder="Ingre...
77     </div>
78
79     <button type="submit" class="btn btn-primary w-100 py-2 mb-3">
80         <i class="fas fa-sign-in-alt"></i> Iniciar Sesión
81     </button>
82 </form>
83
84 <div class="text-center">
85     <a href="?action=forgot" class="text-decoration-none small">
86         <i class="fas fa-key"></i> ¿Olvidó su contraseña?
87     </a>
88 </div>
89
90 <hr class="my-4">
91
92 <div class="text-center">
93     <small class="text-muted">
94         <i class="fas fa-info-circle"></i> Usuario de prueba: <strong>admin</strong> / &...
95     </small>
96 </div>
97 </div>
98
99     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
100 </body>
101 </html>
```

## 60. register.php

Ruta: 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>
32     </div>
33 </body>
34 </html>
```

## 61. reset.php

Ruta: 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: 100%; }
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>
```



## 62. index.php

Ruta: ArchImp/views/backup/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>Respaldos - EcuAsist</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9      <style>
10         body { padding-top: 70px; background: #f4f4f4; }
11         .container { max-width: 1200px; margin: 30px auto; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot...
13     </style>
14 </head>
15 <body>
16     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
17
18     <div class="container">
19         <div class="card">
20             <div class="d-flex justify-content-between align-items-center mb-4">
21                 <h2 class="mb-0">■ Respaldos de Base de Datos</h2>
22                 <button onclick="createBackup()" class="btn btn-primary">
23                     ■ Crear Respaldo Ahora
24                 </button>
25             </div>
26
27             <?php if(isset($_GET['success'])): ?>
28                 <div class="alert alert-success alert-dismissible fade show">
29                     ✓ Respaldo creado correctamente: <strong><?php echo htmlspecialchars($_GET['file']); ?></strong></div>
30                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
31             </div>
32             <?php endif; ?>
33
34             <?php if(isset($_GET['error'])): ?>
35                 <div class="alert alert-danger alert-dismissible fade show">
36                     ✗ Error al crear el respaldo. Verifique la configuración de MySQL.
37                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
38             </div>
39             <?php endif; ?>
40
41             <?php if(isset($_GET['cleanup'])): ?>
42                 <div class="alert alert-info alert-dismissible fade show">
43                     ■ <?php echo (int)$_GET['cleanup']; ?> respaldo(s) antiguo(s) eliminado(s).
44                     <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
45                 </div>
46             <?php endif; ?>
47
48             <?php if(isset($_GET['deleted'])): ?>
49                 <div class="alert alert-success alert-dismissible fade show">
50                     ✓ Respaldo eliminado correctamente.
51                     <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
52                 </div>
53             <?php endif; ?>
54
55             <div class="alert alert-info">
56                 <strong>■ Información:</strong><br>
57                 • Los respaldos se crean automáticamente en formato SQL<br>
58                 • Se recomienda descargar y guardar en ubicación segura<br>
59                 • Los respaldos antiguos (>30 días) pueden eliminarse para liberar espacio
60             </div>
61
62             <div class="d-flex justify-content-between mb-3">
63                 <h4>Respaldos Disponibles (<?php echo count($backups); ?>)</h4>
64                 <?php if(count($backups) > 0): ?>
65                     <button onclick="confirmCleanup()" class="btn btn-warning btn-sm">
66                         ■ Limpiar Antiguos (>30 días)
67                     </button>
68                 </div>
69             </div>
70
71             <?php if(empty($backups)): ?>
72                 <div class="alert alert-secondary text-center">
73                     ■ No hay respaldos disponibles. Crea uno usando el botón superior.
74                 </div>
75             <?php else: ?>
```

```

76         <div class="table-responsive">
77             <table class="table table-hover">
78                 <thead class="table-light">
79                     <tr>
80                         <th>#</th>
81                         <th>Nombre del Archivo</th>
82                         <th>Fecha de Creación</th>
83                         <th>Tamaño</th>
84                         <th>Acciones</th>
85                     </tr>
86                 </thead>
87                 <tbody>
88                     <?php
89                         $counter = 1;
90                         foreach($backups as $backup):
91                             ?>
92                             <tr>
93                                 <td>#<td>?> $counter++ ?></td>
94                                 <td>
95                                     <strong>#<td>?> htmlspecialchars($backup['filename']) ?></strong>
96                                 </td>
97                                 <td>#<td>?> $backup['date'] ?></td>
98                                 <td>
99                                     <span class="badge bg-secondary">#<td>?> $backup['size'] ?></span>
100                                 </td>
101                                 <td>
102                                     <a href="?action=download_backup&file=#<td>?> urlencode($backup['filename']) ?...
103                                     class="btn btn-sm btn-success">
104                                         ■ Descargar
105                                     </a>
106                                     <button onclick="confirmDelete('#<td>?> addslashes($backup['filename']) ?>')"
107                                     class="btn btn-sm btn-danger">
108                                         ■ Eliminar
109                                     </button>
110                                 </td>
111                             </tr>
112                         <?php endforeach; ?>
113                 </tbody>
114             </table>
115         </div>
116     <?php endif; ?>
117 </div>
118 </div>
119
120 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
121 <script>
122     function createBackup() {
123         const modal = document.createElement('div');
124         modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0...
125
126         const modalContent = document.createElement('div');
127         modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh...
128
129         modalContent.innerHTML = `
130             <h3 style="margin: 0 0 15px 0; color: #007bff">■ Crear Respaldo</h3>
131             <p style="margin: 0 0 20px 0; color: #666">
132                 ¿Crear un nuevo respaldo de la base de datos?
133             </p>
134             <p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #e7f3ff; padding: 10px; borde...
135                 <strong>■ Nota:</strong> Este proceso puede tardar unos segundos dependiendo del tamaño...
136             </p>
137             <div style="display: flex; gap: 10px; justify-content: flex-end">
138                 <button type="button" id="cancelCreateBtn" style="padding: 10px 20px; background: #6c757d; color:...
139                     Cancelar
140                 </button>
141                 <button type="button" id="confirmCreateBtn" style="padding: 10px 20px; background: #007bff; color:...
142                     Sí, Crear
143                 </button>
144             </div>
145         `;
146
147         modal.appendChild(modalContent);
148         document.body.appendChild(modal);
149
150         document.getElementById('confirmCreateBtn').onclick = function() {
151             document.body.removeChild(modal);
152             window.location.href = '?action=create_backup';
153         };
154     }

```

```

155     document.getElementById('cancelCreateBtn').onclick = function() {
156         document.body.removeChild(modal);
157     };
158 }
159
160 function confirmCleanup() {
161
162     const modal = document.createElement('div');
163     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0...
164
165     const modalContent = document.createElement('div');
166     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh...
167
168     modalContent.innerHTML = `
169         <h3 style="margin: 0 0 15px 0; color: #ffc107;">■ Limpiar RespalDOS AntiguOS</h3>
170         <p style="margin: 0 0 20px 0; color: #666;">
171             ¿Está seguro de eliminar todos los respaldos con más de <strong>30 días</strong> de anti...
172         </p>
173         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #fff3cd; padding: 10px; borde...
174             <strong>■ Advertencia:</strong> Esta acción no se puede deshacer.
175             Se recomienda descargar los respaldos importantes antes de eliminarlos.
176         </p>
177         <div style="display: flex; gap: 10px; justify-content: flex-end;">
178             <button type="button" id="cancelCleanupBtn" style="padding: 10px 20px; background: #6c757d; color...
179                 Cancelar
180             </button>
181             <button type="button" id="confirmCleanupBtn" style="padding: 10px 20px; background: #ffc107; colo...
182                 Sí, Limpiar
183             </button>
184         </div>
185     `;
186
187     modal.appendChild(modalContent);
188     document.body.appendChild(modal);
189
190     document.getElementById('confirmCleanupBtn').onclick = function() {
191         document.body.removeChild(modal);
192         window.location.href = '?action=cleanup_backups';
193     };
194
195     document.getElementById('cancelCleanupBtn').onclick = function() {
196         document.body.removeChild(modal);
197     };
198 }
199
200 function confirmDelete(filename) {
201     const modal = document.createElement('div');
202     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0...
203
204     const modalContent = document.createElement('div');
205     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh...
206
207     modalContent.innerHTML = `
208         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar RespalDO</h3>
209         <p style="margin: 0 0 20px 0; color: #666;">
210             ¿Está seguro de eliminar el respaldo <strong>${filename}</strong>?
211         </p>
212         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #f8d7da; padding: 10px; borde...
213             <strong>■ Advertencia:</strong> Esta acción no se puede deshacer.
214         </p>
215         <div style="display: flex; gap: 10px; justify-content: flex-end;">
216             <button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color:...
217                 Cancelar
218             </button>
219             <button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color...
220                 Sí, Eliminar
221             </button>
222         </div>
223     `;
224
225     modal.appendChild(modalContent);
226     document.body.appendChild(modal);
227
228     document.getElementById('confirmDeleteBtn').onclick = function() {
229         document.body.removeChild(modal);
230         window.location.href = '?action=delete_backup&file=' + encodeURIComponent(filename);
231     };
232
233     document.getElementById('cancelDeleteBtn').onclick = function() {
234         document.body.removeChild(modal);
235     };
236 }

```

```
234         };
235     }
236     </script>;
237 </body>;
238 </html>;
```

## 63. index.php

Ruta: 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     <style>
9         * { margin: 0; padding: 0; box-sizing: border-box; }
10        body { font-family: Arial, sans-serif; background: #f4f4f4; }
11        .navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; ...
12        .navbar h1 { font-size: 24px; }
13        .navbar .user-info { display: flex; align-items: center; gap: 15px; }
14        .navbar a { color: white; text-decoration: none; padding: 8px 15px; border-radius: 4px; transition: background 0...
15        .navbar a:hover { background: rgba(255,255,255,0.2); }
16        .container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
17        .welcome-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 30px; bord...
18        .welcome-card h2 { margin-bottom: 10px; font-size: 28px; }
19        .badge { display: inline-block; background: rgba(255,255,255,0.3); color: white; padding: 5px 10px; border-radiu...
20
21        .stats-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px; margin-bott...
22        .stat-card { background: white; padding: 25px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); text-...
23        .stat-card:hover { transform: translateY(-5px); box-shadow: 0 4px 20px rgba(0,0,0,0.15); }
24        .stat-number { font-size: 36px; font-weight: bold; color: #007bff; margin-bottom: 5px; }
25        .stat-label { color: #666; font-size: 14px; }
26        .stat-card.green .stat-number { color: #28a745; }
27        .stat-card.red .stat-number { color: #dc3545; }
28        .stat-card.orange .stat-number { color: #fd7e14; }
29        .stat-card.purple .stat-number { color: #6f42c1; }
30
31        .menu-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(280px, 1fr)); gap: 20px; }
32        .menu-item { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); curso...
33        .menu-item:hover { transform: translateY(-5px); box-shadow: 0 4px 15px rgba(0,0,0,0.15); }
34        .menu-item h3 { color: #007bff; margin-bottom: 10px; font-size: 18px; }
35        .menu-item p { color: #666; font-size: 14px; }
36        .menu-item.green { border-left-color: #28a745; }
37        .menu-item.green h3 { color: #28a745; }
38        .menu-item.orange { border-left-color: #fd7e14; }
39        .menu-item.orange h3 { color: #fd7e14; }
40        .menu-item.purple { border-left-color: #6f42c1; }
41        .menu-item.purple h3 { color: #6f42c1; }
42    </style>
43 </head>
44 <body>
45     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
46
47     <div class="container">
48         <div class="welcome-card">
49             <h2>Bienvenido, <?php echo $_SESSION['first_name']; . ' ' . $_SESSION['last_name']; ?></h2>
50             <p>Roles asignados:
51                 <?php foreach($_SESSION['roles'] as $role): ?>
52                     <span class="badge"><?php echo ucfirst($role); ?></span>
53                 <?php endforeach; ?>
54             </p>
55         </div>
56
57         <?php if(Security::hasRole('autoridad') && isset($stats['total_students'])): ?>
58             <div class="stats-grid">
59                 <div class="stat-card">
60                     <div class="stat-number"><?php echo $stats['total_students']; ?></div>
61                     <div class="stat-label">Estudiantes</div>
62                 </div>
63                 <div class="stat-card green">
64                     <div class="stat-number"><?php echo $stats['total_teachers']; ?></div>
65                     <div class="stat-label">Docentes</div>
66                 </div>
67                 <div class="stat-card orange">
68                     <div class="stat-number"><?php echo $stats['total_courses']; ?></div>
69                     <div class="stat-label">Cursos</div>
70                 </div>
71                 <?php if($stats['today_attendance']['total'] > 0): ?>
72                     <div class="stat-card purple">
73                         <div class="stat-number">
74                             <?php echo round(($stats['today_attendance']['presente'] / $stats['today_attendance']['total']) * 100, 1...
75                         </div>
```

```

76         <div class="stat-label">Asistencia Hoy</div>
77     </div>
78     <?php endif; ?>
79 </div>
80 <?php endif; ?>
81
82 <?php if(Security::hasRole('docente') && isset($stats['my_courses'])): ?>
83 <div class="stats-grid">
84     <div class="stat-card">
85         <div class="stat-number"><?= $stats['my_courses'] ?></div>
86         <div class="stat-label">Mis Cursos</div>
87     </div>
88     <div class="stat-card green">
89         <div class="stat-number"><?= $stats['my_students'] ?></div>
90         <div class="stat-label">Mis Estudiantes</div>
91     </div>
92     <?php if(!empty($stats['tutor_course'])): ?>
93     <div class="stat-card purple" style="grid-column: span 1;">
94         <div style="font-size: 28px; margin-bottom: 6px;">■</div>
95         <div style="font-size: 13px; font-weight: bold; color: #6f42c1; margin-bottom: 4px;">
96             Soy Tutor de:
97         </div>
98         <div style="font-size: 14px; color: #333; font-weight: bold;">
99             <?= htmlspecialchars($stats['tutor_course']) ?>
100         </div>
101     </div>
102     <?php endif; ?>
103 </div>
104 <?php endif; ?>
105
106 <?php if(Security::hasRole('estudiante') && isset($stats['my_attendance'])): ?>
107 <div class="stats-grid">
108     <div class="stat-card green">
109         <div class="stat-number">
110             <?= $stats['my_attendance']['total'] > 0 ? round(($stats['my_attendance']['presente'] / $stats...
111         </div>
112         <div class="stat-label">Mi Asistencia (30 días)</div>
113     </div>
114     <div class="stat-card red">
115         <div class="stat-number"><?= $stats['my_attendance']['ausente'] ?></div>
116         <div class="stat-label">Ausencias</div>
117     </div>
118     <div class="stat-card orange">
119         <div class="stat-number"><?= $stats['my_attendance']['tardanza'] ?></div>
120         <div class="stat-label">Tardanzas</div>
121     </div>
122 </div>
123 <?php endif; ?>
124
125 <?php if(Security::hasRole('representante') && isset($stats['my_children'])): ?>
126 <div class="stats-grid">
127     <div class="stat-card">
128         <div class="stat-number"><?= $stats['my_children'] ?></div>
129         <div class="stat-label">Representados</div>
130     </div>
131 </div>
132 <?php endif; ?>
133
134 <div class="menu-grid">
135     <?php if(Security::hasRole(['docente', 'autoridad'])): ?>
136     <div class="menu-item green" onclick="location.href='?action=attendance_register'">
137         <h3>■ Registrar Asistencia</h3>
138         <p>Tomar asistencia de estudiantes</p>
139     </div>
140     <?php endif; ?>
141
142     <?php if(Security::hasRole(['docente', 'inspector', 'autoridad'])): ?>
143     <div class="menu-item purple" onclick="location.href='?action=attendance_view'">
144         <h3>■ Ver Asistencias</h3>
145         <p>Consultar registros de asistencia</p>
146     </div>
147     <?php endif; ?>
148
149     <?php if(Security::hasRole('autoridad')): ?>
150     <div class="menu-item purple" onclick="location.href='?action=stats'">
151         <h3>■ Estadísticas</h3>
152         <p>Análisis y métricas de asistencia</p>
153     </div>
154     <?php endif; ?>

```

```

155
156 <?php if(Security::hasRole(['docente', 'autoridad'])): ?>
157 <div class="menu-item orange" onclick="location.href='?action=attendance_calendar'">
158 <h3>■ Calendario</h3>
159 <p>Vista mensual de asistencias</p>
160 </div>

161 <?php endif; ?>
162
163 <?php if(Security::hasRole('estudiante')): ?>
164 <div class="menu-item" onclick="location.href='?action=my_attendance'">
165 <h3>■ Mi Asistencia</h3>
166 <p>Ver mi historial de asistencia</p>
167 </div>
168 <?php endif; ?>
169
170 <?php if(Security::hasRole('estudiante')): ?>
171 <div class="menu-item green" onclick="location.href='?action=my_justifications'">
172 <h3>■ Mis Justificaciones</h3>
173 <p>Justificar ausencias</p>
174 </div>
175 <?php endif; ?>
176
177 <?php if(Security::hasRole(['autoridad', 'inspector'])): ?>
178 <div class="menu-item orange" onclick="location.href='?action=pending_justifications'">
179 <h3>■ Revisar Justificaciones</h3>
180 <p>Aprobar o rechazar justificaciones</p>
181 </div>
182 <?php endif; ?>
183
184 <?php if(Security::hasRole('autoridad')): ?>
185 <div class="menu-item orange" onclick="location.href='?action=users'">
186 <h3>■ Gestión de Usuarios</h3>
187 <p>Administrar usuarios y roles</p>
188 </div>
189
190 <?php if(Security::hasRole('autoridad')): ?>
191 <div class="menu-item purple" onclick="location.href='?action=backups'">
192 <h3>■ RespalDOS</h3>
193 <p>Gestionar respaldos de base de datos</p>
194 </div>
195 <?php endif; ?>
196
197 <div class="menu-item purple" onclick="location.href='?action=reports'">
198 <h3>■ Reportes</h3>
199 <p>Generar informes PDF y Excel</p>
200 </div>
201
202 <div class="menu-item" onclick="location.href='?action=academic'">
203 <h3>■ Configuración Académica</h3>
204 <p>Cursos, materias, matrículas</p>
205 </div>
206
207 <div class="menu-item green" onclick="location.href='?action=assignments'">
208 <h3>■ Asignaciones Docentes</h3>
209 <p>Gestionar docentes y tutores</p>
210 </div>
211
212 <div class="menu-item orange" onclick="location.href='?action=manage_representatives'">
213 <h3>■ Gestión Representantes</h3>
214 <p>Asignar representantes a estudiantes</p>
215 </div>
216 <?php endif; ?>
217
218 <?php if(Security::hasRole('representante')): ?>
219 <div class="menu-item purple" onclick="location.href='?action=my_children'">
220 <h3>■ Mis Representados</h3>
221 <p>Ver estudiantes y su asistencia</p>
222 </div>
223 <?php endif; ?>
224 </div>
225 </div>
226 </body>
227 </html>

```

## 64. index.php

Ruta: ArchImp/views/institution/index.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Configuración de Institución - 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        label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
14        input, select, textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
15        button { padding: 10px 20px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointe...
16        button:hover { background: #0056b3; }
17        .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
18        .grid { display: grid; grid-template-columns: 1fr 1fr; gap: 15px; }
19        .grid-3 { display: grid; grid-template-columns: 2fr 1fr; gap: 15px; }
20        h2 { margin-bottom: 20px; color: #333; }
21        .badge {
22            display: inline-block;
23            background: #28a745;
24            color: white;
25            padding: 5px 10px;
26            border-radius: 3px;
27            font-size: 12px;
28            margin: 3px;
29        }
30        .btn-remove {
31            background: none;
32            border: none;
33            color: white;
34            font-size: 16px;
35            font-weight: bold;
36            cursor: pointer;
37            padding: 0;
38            margin-left: 5px;
39        }
40        .btn-remove:hover { color: #ff0000; }
41    </style>
42 </head>
43 <body>
44     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
45
46     <div class="container">
47         <?php if(isset($_GET['success'])): ?>
48             <div class="success">✓ Información actualizada correctamente</div>
49         <?php endif; ?>
50         <?php if(isset($_GET['shift_assigned'])): ?>
51             <div class="success">✓ Jornada asignada correctamente</div>
52         <?php endif; ?>
53         <?php if(isset($_GET['shift_removed'])): ?>
54             <div class="success">✓ Jornada eliminada correctamente</div>
55         <?php endif; ?>
56
57         <div class="card">
58             <h2>■ Información de la Institución</h2>
59             <form method="POST" action="?action=update_institution" enctype="multipart/form-data">
60                 <div class="grid-3">
61                     <div class="form-group">
62                         <label>Nombre de la Institución </label>
63                         <input type="text" name="name" value="<?= $institution['name'] ?>" required>
64                     </div>
65
66                     <div class="form-group">
67                         <label>Logo de la Institución</label>
68                         <?php if(isset($institution['logo_path'])) &amp;&amp; $institution['logo_path']): ?>
69                             
71                         <input type="file" name="logo" accept="image/png,image/jpeg,image/jpg">
72                         <small style="color: #666; display: block; margin-top: 5px;">PNG, JPG - 300x300px recomend...
73                     </div>
74                 </div>
75             </form>
76         </div>
77     </div>
```



```

76     <div class="grid">
77         <div class="form-group">
78             <label>Código AMIE</label>
79             <input type="text" name="amie_code" value="&lt;?= $institution['amie_code'] ?? ' ' ?&gt;" plac...
80         </div>
81
82     <div class="form-group">
83         <label>Nombre del Director/Rector</label>
84         <input type="text" name="director_name" value="&lt;?= $institution['director_name'] ?? ' ' ?&g...
85     </div>
86
87     <div class="form-group">
88         <label>Provincia *</label>
89         <select name="province" id="province" required>
90             <option value="">Seleccionar provincia...</option>
91             <option value="Pichincha" &lt;?= ($institution['province'] ?? ' ') == 'Pichincha' ? 'selec...
92             <option value="Guayas" &lt;?= ($institution['province'] ?? ' ') == 'Guayas' ? 'selected' : ...
93             <option value="Azuay" &lt;?= ($institution['province'] ?? ' ') == 'Azuay' ? 'selected' : ...
94             <option value="Manabí" &lt;?= ($institution['province'] ?? ' ') == 'Manabí' ? 'selected' : ...
95             <option value="El Oro" &lt;?= ($institution['province'] ?? ' ') == 'El Oro' ? 'selected' : ...
96             <option value="Los Ríos" &lt;?= ($institution['province'] ?? ' ') == 'Los Ríos' ? 'sele...
97             <option value="Tungurahua" &lt;?= ($institution['province'] ?? ' ') == 'Tungurahua' ? 'sel...
98             <option value="Esmeraldas" &lt;?= ($institution['province'] ?? ' ') == 'Esmeraldas' ? 'sel...
99             <option value="Chimborazo" &lt;?= ($institution['province'] ?? ' ') == 'Chimborazo' ? 'sel...
100            <option value="Imbabura" &lt;?= ($institution['province'] ?? ' ') == 'Imbabura' ? 'sele...
101            <option value="Cotopaxi" &lt;?= ($institution['province'] ?? ' ') == 'Cotopaxi' ? 'sele...
102            <option value="Santo Domingo de los Tsáchilas" &lt;?= ($institution['province'] ?? ' ') == ...
103            <option value="Santa Elena" &lt;?= ($institution['province'] ?? ' ') == 'Santa Elena' ? 's...
104            <option value="Loja" &lt;?= ($institution['province'] ?? ' ') == 'Loja' ? 'selected' : ' ' ...
105            <option value="Cañar" &lt;?= ($institution['province'] ?? ' ') == 'Cañar' ? 'selected' : '...
106            <option value="Sucumbíos" &lt;?= ($institution['province'] ?? ' ') == 'Sucumbíos' ? 'selec...
107            <option value="Orellana" &lt;?= ($institution['province'] ?? ' ') == 'Orellana' ? 'sele...
108            <option value="Carchi" &lt;?= ($institution['province'] ?? ' ') == 'Carchi' ? 'selected' : ...
109            <option value="Bolívar" &lt;?= ($institution['province'] ?? ' ') == 'Bolívar' ? 'selected'...
110            <option value="Napo" &lt;?= ($institution['province'] ?? ' ') == 'Napo' ? 'selected' : ' ' ...
111            <option value="Pastaza" &lt;?= ($institution['province'] ?? ' ') == 'Pastaza' ? 'selected'...
112            <option value="Morona Santiago" &lt;?= ($institution['province'] ?? ' ') == 'Morona Santia...
113            <option value="Zamora Chinchipe" &lt;?= ($institution['province'] ?? ' ') == 'Zamora Chinc...
114            <option value="Galápagos" &lt;?= ($institution['province'] ?? ' ') == 'Galápagos' ? 'selec...
115        </select>
116    </div>
117
118    <div class="form-group">
119        <label>Ciudad *</label>
120        <select name="city" id="city" required>
121            <option value="">Primero seleccione provincia...</option>
122        </select>
123    </div>
124 </div>
125
126 <div class="form-group">
127     <label>Dirección *</label>
128     <input type="text" name="address" value="&lt;?= $institution['address'] ?&gt;" required>
129 </div>
130
131 <div class="grid">
132     <div class="form-group">
133         <label>Teléfono</label>
134         <input type="tel" name="phone" value="&lt;?= $institution['phone'] ?? ' ' ?&gt;"&gt;
135     </div>
136
137     <div class="form-group">
138         <label>Email *</label>
139         <input type="email" name="email" value="&lt;?= $institution['email'] ?&gt;" required>
140     </div>
141
142     <div class="form-group">
143         <label>Sitio Web</label>
144         <input type="text" name="website" id="website" value="&lt;?= $institution['website'] ?? ' ' ?&...
145     </div>
146 </div>
147
148 <button type="submit">Guardar Cambios</button>
149 </form>
150 </div>
151
152 <div class="card">
153     <h2>■ Jornadas de la Institución</h2>
154

```

```

155         <div style="margin-bottom: 20px;">
156             <strong>Jornadas Activas:</strong><br>
157             <?php
158                 $assignedShifts = $this->institutionShiftModel->getByInstitution(1);
159                 if(empty($assignedShifts)):
160                     ?>
161
162                     <em style="color: #999;">No hay jornadas asignadas</em>
163                 <?php else: ?>
164                     <?php foreach($assignedShifts as $shift): ?>
165                         <span class="badge">
166                             <?php ucfirst($shift['name']) ?>
167                             <form method="POST" action="?action=remove_institution_shift" style="display: inline;" on...
168                             <input type="hidden" name="shift_id" value="<?php $shift['id'] ?>">
169                             <button type="submit" class="btn-remove">x</button>
170                         </form>
171                     </span>
172                 <?php endforeach: ?>
173             </div>
174
175             <form method="POST" action="?action=assign_institution_shift" style="display: flex; gap: 10px; align-item...
176             <div style="flex: 1;">
177                 <label>Asignar Nueva Jornada</label>
178                 <select name="shift_id" required>
179                     <option value="">Seleccionar...</option>
180                     <?php foreach($allShifts as $shift): ?>
181                         <?php if(!in_array($shift['id'], $assignedShiftIds)): ?>
182                             <option value="<?php $shift['id'] ?>"><?php ucfirst($shift['name']) ?></option>
183                         <?php endif: ?>
184                     <?php endforeach: ?>
185                 </select>
186             </div>
187             <button type="submit">Asignar Jornada</button>
188         </div>
189     </div>
190 </div>
191
192 <script>
193 const cities = {
194     'Pichincha': ['Quito', 'Cayambe', 'Mejía', 'Pedro Moncayo', 'Rumiñahui', 'San Miguel de los Bancos', 'Pedro Vice...
195     'Guayas': ['Guayaquil', 'Durán', 'Milagro', 'Daule', 'Samborondón', 'Balao', 'Balzar', 'Colimes', 'El Empalme', ...
196     'Azuay': ['Cuenca', 'Gualaceo', 'Paute', 'Sigsig', 'Girón', 'Santa Isabel', 'Nabón', 'Oña', 'Pucará', 'San Ferna...
197     'Manabí': ['Portoviejo', 'Manta', 'Chone', 'El Carmen', 'Flavio Alfaro', 'Jipijapa', 'Junín', 'Montecristi', 'Pa...
198     'El Oro': ['Machala', 'Pasaje', 'Santa Rosa', 'Huaquillas', 'Arenillas', 'Atahualpa', 'Balsas', 'Chilla', 'El Gu...
199     'Los Ríos': ['Babahoyo', 'Quevedo', 'Baba', 'Montalvo', 'Pueblo Viejo', 'Urdaneta', 'Ventanas', 'Vinces', 'Palenq...
200     'Tungurahua': ['Ambato', 'Baños de Agua Santa', 'Pelileo', 'Píllaro', 'Cevallos', 'Mocha', 'Patate', 'Quero', 'T...
201     'Esmeraldas': ['Esmeraldas', 'Quinindé', 'Atacames', 'Eloy Alfaro', 'Muisne', 'San Lorenzo', 'Rioverde', 'La Con...
202     'Chimborazo': ['Riobamba', 'Alausí', 'Guano', 'Colta', 'Chambo', 'Chunchi', 'Guamote', 'Pallatanga', 'Penipe', '...
203     'Imbabura': ['Ibarra', 'Otavalo', 'Cotacachi', 'Antonio Ante', 'Pimampiro', 'San Miguel de Urcuquí'],
204     'Cotopaxi': ['Latacunga', 'La Maná', 'Pujilí', 'Salcedo', 'Saquisilí', 'Pangua', 'Sigchos'],
205     'Santo Domingo de los Tsáchilas': ['Santo Domingo'],
206     'Santa Elena': ['La Libertad', 'Salinas', 'Santa Elena'],
207     'Loja': ['Loja', 'Catamayo', 'Macará', 'Saraguro', 'Calvas', 'Celica', 'Chaguarpamba', 'Espíndola', 'Gonzanamá',...
208     'Cañar': ['Azogues', 'Cañar', 'La Troncal', 'Biblián', 'El Tambo', 'Dégel', 'Suscal'],
209     'Sucumbios': ['Nueva Loja (Lago Agrio)', 'Shushufindi', 'Gonzalo Pizarro', 'Putumayo', 'Sucumbios', 'Cascales', ...
210     'Orellana': ['Francisco de Orellana (Coca)', 'La Joya de los Sachas', 'Aguarico', 'Loreto'],
211     'Carchi': ['Tulcán', 'Montúfar', 'Bolívar', 'Espejo', 'Mira', 'San Pedro de Huaca'],
212     'Bolívar': ['Guaranda', 'San Miguel', 'Chillanes', 'Chimbo', 'Echeandía', 'Caluma', 'Las Naves'],
213     'Napo': ['Tena', 'Archidona', 'El Chaco', 'Quijos', 'Carlos Julio Arosemena Tola'],
214     'Pastaza': ['Puyo', 'Mera', 'Arajuno', 'Santa Clara'],
215     'Morona Santiago': ['Macas', 'Gualaquiza', 'Sucúa', 'Limón Indanza', 'Palora', 'Santiago', 'Huamboya', 'San Juan...
216     'Zamora Chinchipe': ['Zamora', 'Yantzaza', 'Chinchipe', 'Nangaritzza', 'Yacuambi', 'El Pangui', 'Centinela del Có...
217     'Galápagos': ['Puerto Baquerizo Moreno', 'Puerto Ayora', 'Puerto Villamil']
218 };
219
220 const currentCity = '<?php $institution['city'] ?? '' ?>';
221 const currentProvince = '<?php $institution['province'] ?? '' ?>';
222
223 document.getElementById('province').addEventListener('change', function() {
224     const province = this.value;
225     const citySelect = document.getElementById('city');
226
227     citySelect.innerHTML = '<option value="">Seleccionar ciudad...</option>';
228
229     if (province && cities[province]) {
230         cities[province].forEach(city => {
231             const option = document.createElement('option');
232             option.value = city;
233             option.textContent = city;

```

```

234         citySelect.appendChild(option);
235     });
236 }
237 });
238
239 // Cargar ciudades al inicio si hay provincia seleccionada
240 if (currentProvince && cities[currentProvince]) {
241     const citySelect = document.getElementById('city');
242     citySelect.innerHTML = '<option value="">Seleccionar ciudad...</option>';
243     cities[currentProvince].forEach(city => {
244         const option = document.createElement('option');
245         option.value = city;
246         option.textContent = city;
247         if (city === currentCity) {
248             option.selected = true;
249         }
250         citySelect.appendChild(option);
251     });
252 }
253
254 // Autocompletar URL
255 document.getElementById('website').addEventListener('blur', function() {
256     let url = this.value.trim();
257
258     if (url && !url.match(/^https?:\/\//)) {
259         if (!url.startsWith('www.')) {
260             url = 'www.' + url;
261         }
262         this.value = 'https://' + url;
263     }
264 });
265
266 function confirmRemoveShift(event, shiftName) {
267     event.preventDefault();
268
269     const modal = document.createElement('div');
270     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...';
271
272     const modalContent = document.createElement('div');
273     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...';
274
275     modalContent.innerHTML = `
276         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Jornada</h3>
277         <p style="margin: 0 0 20px 0; color: #666;">
278             ¿Está seguro de eliminar la jornada <strong>${shiftName}</strong> de esta institución?
279         </p>
280         <div style="display: flex; gap: 10px; justify-content: flex-end;">
281             <button type="button" id="cancelBtn" style="padding: 10px 20px; background: #6c757d; color: white; bo...
282                 Cancelar
283             </button>
284             <button type="button" id="confirmBtn" style="padding: 10px 20px; background: #dc3545; color: white; b...
285                 Sí, Eliminar
286             </button>
287         </div>
288     `;
289
290     modal.appendChild(modalContent);
291     document.body.appendChild(modal);
292
293     const form = event.target;
294
295     document.getElementById('confirmBtn').onclick = function() {
296         document.body.removeChild(modal);
297         form.submit();
298     };
299
300     document.getElementById('cancelBtn').onclick = function() {
301         document.body.removeChild(modal);
302     };
303
304     return false;
305 }
306 </script>
307 </body>
308 </html>

```

## 65. my\_list.php

Ruta: 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      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9      <style>
10         body { padding-top: 70px; background: #f4f4f4; }
11         .container { max-width: 1200px; margin: 30px auto; }
12         .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot...
13         .badge-pending { background: #ffc107; color: #000; padding: 6px 12px; border-radius: 12px; font-size: 12px; font...
14         .badge-approved { background: #28a745; color: white; padding: 6px 12px; border-radius: 12px; font-size: 12px; fo...
15         .badge-rejected { background: #dc3545; color: white; padding: 6px 12px; border-radius: 12px; font-size: 12px; fo...
16         .table th { background: #f8f9fa; font-weight: bold; }
17         .empty-state {
18             text-align: center;
19             padding: 60px 20px;
20             color: #999;
21         }
22         .empty-state .icon {
23             font-size: 64px;
24             margin-bottom: 20px;
25             opacity: 0.5;
26         }
27     </style>
28 </head>
29 <body>
30     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
31
32     <div class="container">
33         <div class="card">
34             <div class="d-flex justify-content-between align-items-center mb-4">
35                 <h2 class="mb-0">■ Mis Justificaciones</h2>
36             </div>
37
38             <?php if(isset($_GET['success'])): ?>
39                 <div class="alert alert-success alert-dismissible fade show">
40                     ✓ Justificación enviada correctamente. Está pendiente de revisión.
41                     <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
42                 </div>
43             <?php endif; ?>
44
45             <?php if(empty($justifications)): ?>
46                 <div class="empty-state">
47                     <div class="icon">■</div>
48                     <h4>No tienes justificaciones enviadas</h4>
49                     <p class="text-muted">
50                         Las justificaciones que envíes aparecerán aquí.<br>
51                         Puedes justificar ausencias desde "Mi Asistencia".
52                     </p>
53                 </div>
54             <?php else: ?>
55                 <div class="table-responsive">
56                     <table class="table table-hover">
57                         <thead>
58                             <tr>
59                                 <th>#</th>
60                                 <th>Fecha</th>
61                                 <th>Motivo</th>
62                                 <th>Documento</th>
63                                 <th>Estado</th>
64                                 <th>Fecha Envío</th>
65                                 <th>Revisado Por</th>
66                                 <th>Observaciones</th>
67                             </tr>
68                         </thead>
69                         <tbody>
70                             <?php
71                                 $counter = 1;
72                                 foreach($justifications as $just):
73                                     ?>
74                                     <tr>
75                                         <td>#<?php echo $counter++ ?></td>
```

```

76         <td><td>?<td>= date('d/m/Y', strtotime($just['attendance_date'])) ?<td></td><td>
77         <td><td>?<td>= htmlspecialchars($just['reason']) ?<td></td><td>
78         <td><td>
79         <td>?<td>php if($just['document_path']): ?<td>
80         <td>a href="<td>?<td>= BASE_URL ?<td></td>?<td>= $just['document_path'] ?<td>";
81         target="_blank"
82         class="btn btn-sm btn-outline-primary"<td>
83         Ver Documento
84         </a><td>
85         <td>?<td>php else: ?<td>
86         <td>?<td>span class="text-muted"<td>Sin documento</span><td>
87         <td>?<td>php endif: ?<td>
88         <td></td><td>
89         <td></td><td>
90         <td>?<td>php if($just['status'] === 'pendiente'): ?<td>
91         <td>?<td>span class="badge-pending"<td>Pendiente</span><td>
92         <td>?<td>php elseif($just['status'] === 'aprobado'): ?<td>
93         <td>?<td>span class="badge-approved"<td>Aprobado</span><td>
94         <td>?<td>php else: ?<td>
95         <td>?<td>span class="badge-rejected"<td>Rechazado</span><td>
96         <td>?<td>php endif: ?<td>
97         <td></td><td>
98         <td><td><td>?<td>= date('d/m/Y H:i', strtotime($just['submitted_at'])) ?<td></td></td><td>
99         <td></td><td>
100        <td>?<td>php if($just['reviewed_by']): ?<td>
101        <td>?<td>= htmlspecialchars($just['reviewer_name']) ?<td>
102        <td><br><td>
103        <td>?<td>small class="text-muted"<td>
104        <td>?<td>= date('d/m/Y H:i', strtotime($just['reviewed_at'])) ?<td>
105        <td></td><td>
106        <td>?<td>php else: ?<td>
107        <td>?<td>span class="text-muted"<td></td></span><td>
108        <td>?<td>php endif: ?<td>
109        <td></td><td>
110        <td></td><td>
111        <td>?<td>php if($just['review_notes']): ?<td>
112        <td>?<td>small<td><td>?<td>= htmlspecialchars($just['review_notes']) ?<td></td></td><td>
113        <td>?<td>php else: ?<td>
114        <td>?<td>span class="text-muted"<td></td></span><td>
115        <td>?<td>php endif: ?<td>
116        <td></td><td>
117        <td></td><td>
118        <td>?<td>php endforeach: ?<td>
119        <td></td><td><td>
120        <td></table><td>
121        <td></div><td>
122
123        <td>!-- Resumen estadístico --<td>
124        <td>div class="row mt-4"<td>
125        <td>div class="col-md-4"<td>
126        <td>div class="card bg-warning bg-opacity-10 border-warning"<td>
127        <td>div class="card-body text-center"<td>
128        <td>h3 class="mb-0"<td>
129        <td>?<td>= count(array_filter($justifications, fn($j) =<td> $j['status'] === 'pendiente'...
130        <td></h3><td>
131        <td>?<td>small class="text-muted"<td>Pendientes</td></td><td>
132        <td></div><td>
133        <td></div><td>
134        <td></div><td>
135        <td>div class="col-md-4"<td>
136        <td>div class="card bg-success bg-opacity-10 border-success"<td>
137        <td>div class="card-body text-center"<td>
138        <td>h3 class="mb-0"<td>
139        <td>?<td>= count(array_filter($justifications, fn($j) =<td> $j['status'] === 'aprobado'...
140        <td></h3><td>
141        <td>?<td>small class="text-muted"<td>Aprobadas</td></td><td>
142        <td></div><td>
143        <td></div><td>
144        <td></div><td>
145        <td>div class="col-md-4"<td>
146        <td>div class="card bg-danger bg-opacity-10 border-danger"<td>
147        <td>div class="card-body text-center"<td>
148        <td>h3 class="mb-0"<td>
149        <td>?<td>= count(array_filter($justifications, fn($j) =<td> $j['status'] === 'rechazado'...
150        <td></h3><td>
151        <td>?<td>small class="text-muted"<td>Rechazadas</td></td><td>
152        <td></div><td>
153        <td></div><td>
154        <td></div><td>

```

```
155         </div>;
156     <?php endif; ?>
157 </div>;
158
159 <div class="text-center">
160     <a href="?action=my_attendance" class="btn btn-secondary">
161         ← Volver a Mi Asistencia
162     </a>;
163 </div>;
164 </div>;
165
166 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>;
167 </body>;
168 </html>;
```

## 66. pending.php

Ruta: ArchImp/views/justifications/pending.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Justificaciones Pendientes - 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-bot...
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        .btn { padding: 6px 12px; border: none; border-radius: 4px; cursor: pointer; font-size: 13px; margin-right: 5px;...
19        .btn-approve { background: #28a745; color: white; }
20        .btn-reject { background: #dc3545; color: white; }
21        .btn-view { background: #007bff; color: white; }
22        .modal { display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5)...
23        .modal-content { background: white; margin: 50px auto; padding: 30px; border-radius: 8px; max-width: 600px; }
24        .modal-close { float: right; font-size: 28px; cursor: pointer; }
25        textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; margin: 10px 0; }
26        .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
27    </style>
28 </head>
29 <body>
30     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
31
32     <div class="container">
33         <?php if(isset($_GET['success'])): ?>
34             <div class="success">✓ Justificación procesada correctamente</div>
35         <?php endif; ?>
36
37         <div class="card">
38             <h2 style="margin-bottom: 20px;">Justificaciones por Revisar (<?= count($justifications) ?>)<!-- ...
39
40             <?php if(count($justifications) > 0): ?>
41                 <table>
42                     <thead>
43                         <tr>
44                             <th>Estudiante</th>
45                             <th>Curso</th>
46                             <th>Fecha Ausencia</th>
47                             <th>Asignatura</th>
48                             <th>Presentado por</th>
49                             <th>Fecha Solicitud</th>
50                             <th>Acciones</th>
51                         </tr>
52                     </thead>
53                     <tbody>
54                         <?php foreach($justifications as $just): ?>
55                             <tr>
56                                 <td><?= $just['student_name'] ?></td>
57                                 <td><?= $just['course_name'] ?? '-' ?></td>
58                                 <td><?= date('d/m/Y', strtotime($just['attendance_date'])) ?></td>
59                                 <td><?= $just['subject_name'] ?></td>
60                                 <td><?= $just['submitted_by_name'] ?></td>
61                                 <td><?= date('d/m/Y H:i', strtotime($just['created_at'])) ?></td>
62                                 <td>
63                                     <button class="btn btn-view" onclick="viewJustification(<?= $just['id'] ?>, '<?=...
64                                 </td>
65                             </tr>
66                         <?php endforeach; ?>
67                     </tbody>
68                 </table>
69                 <?php else: ?>
70                     <p style="color: #666; text-align: center; padding: 40px;">No hay justificaciones pendientes</p>
71                 <?php endif; ?>
72             </div>
73     </div>
74
75     <!-- Modal -->
```

```

76     <div id="modal" class="modal">
77         <div class="modal-content">
78             <span class="modal-close" onclick="closeModal()">&times;</span>
79             <h3>Revisar Justificación</h3>
80             <div id="modal-body"></div>
81
82             <form method="POST" action="?action=review_justification">
83                 <input type="hidden" name="justification_id" id="justification_id">
84                 <textarea name="notes" placeholder="Notas de revisión (opcional)"></textarea>
85
86                 <div style="margin-top: 15px;">
87                     <button type="submit" name="review_action" value="approve" class="btn btn-approve">✓ Aprobar&l...
88                     <button type="submit" name="review_action" value="reject" class="btn btn-reject">✗ Rechazar&l...
89                 </div>
90             </form>
91         </div>
92     </div>
93
94     <script>
95         function viewJustification(id, reason, docPath) {
96             const justificationIdInput = window.document.querySelector('#justification_id');
97             const modalBody = window.document.querySelector('#modal-body');
98             const modal = window.document.querySelector('#modal');
99
100             if (!justificationIdInput || !modalBody || !modal) {
101                 console.error('Elementos del modal no encontrados');
102                 return;
103             }
104
105             justificationIdInput.value = id;
106
107             let html = '<p><strong>Motivo:</strong></p>' + reason + '</p>';
108
109             if (docPath && docPath.length > 0) {
110                 html += '<p style="margin-top: 15px;"><strong>Documento:</strong> <a href="<? B...
111             } else {
112                 html += '<p style="margin-top: 15px; color: #999;">&em>Sin documento adjunto&em></p>...
113             }
114
115             modalBody.innerHTML = html;
116             modal.style.display = 'block';
117         }
118
119         function closeModal() {
120             const modal = window.document.querySelector('#modal');
121             if (modal) {
122                 modal.style.display = 'none';
123             }
124         }
125
126         window.onclick = function(event) {
127             const modal = window.document.querySelector('#modal');
128             if (modal && event.target == modal) {
129                 closeModal();
130             }
131         }
132     </script>
133 </body>
134 </html>

```



## 67. reviewed.php

Ruta: ArchImp/views/justifications/reviewed.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Justificaciones Revisadas - 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-bot...
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        .btn { padding: 6px 12px; border: none; border-radius: 4px; cursor: pointer; font-size: 13px; margin-right: 5px;...
19        .btn-primary { background: #007bff; color: white; }
20        .btn-success { background: #28a745; color: white; }
21        .btn-danger { background: #dc3545; color: white; }
22        .btn-outline { background: white; color: #007bff; border: 1px solid #007bff; }
23        .badge { padding: 4px 10px; border-radius: 12px; font-size: 12px; font-weight: bold; }
24        .badge-success { background: #d4edda; color: #155724; }
25        .badge-danger { background: #f8d7da; color: #721c24; }
26        .filter-bar { margin-bottom: 20px; display: flex; gap: 10px; }
27        .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
28        .modal { display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5)...
29        .modal-content { background: white; margin: 50px auto; padding: 30px; border-radius: 8px; max-width: 600px; }
30        .modal-close { float: right; font-size: 28px; cursor: pointer; }
31    </style>
32 </head>
33 <body>
34     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
35
36     <div class="container">
37         <div class="card">
38             <h2 style="margin-bottom: 20px;">■ Justificaciones Revisadas</h2>
39
40             <!-- Filtros -->
41             <div class="filter-bar">
42                 <a href="?action=reviewed_justifications"
43                     class="btn <?=( $filter === 'all' ) ? 'btn-primary' : 'btn-outline' ?>">
44                     Todas (<?=( count($justifications) ?>
45                 </a>
46                 <a href="?action=reviewed_justifications&filter=aprobado"
47                     class="btn <?=( $filter === 'aprobado' ) ? 'btn-success' : 'btn-outline' ?>">
48                     ■ Aprobadas
49                 </a>
50                 <a href="?action=reviewed_justifications&filter=rechazado"
51                     class="btn <?=( $filter === 'rechazado' ) ? 'btn-danger' : 'btn-outline' ?>">
52                     ■ Rechazadas
53                 </a>
44             </div>
45
46             <?php if (count($justifications) > 0): ?>
47                 <table>
48                     <thead>
49                         <tr>
50                             <th>#</th>
51                             <th>Estudiante</th>
52                             <th>Curso</th>
53                             <th>Asignatura</th>
54                             <th>Fecha Falta</th>
55                             <th>Enviado por</th>
56                             <th>Estado</th>
57                             <th>Revisado por</th>
58                             <th>Acciones</th>
59                         </tr>
60                     </thead>
61                     <tbody>
62                         <?php $i = 1; foreach ($justifications as $j): ?>
63                             <tr>
64                                 <td><?=( $i++ ?></td>
65                                 <td><?=( htmlspecialchars($j['student_name']) ?></td>
```

```

76         &td&td&td;?<= htmlspecialchars($j['course_name'] ?? '-') ?<td&td&td;
77         &td&td&td;?<= htmlspecialchars($j['subject_name']) ?<td&td&td;
78         &td&td&td;?<= date('d/m/Y', strtotime($j['attendance_date'])) ?<td&td&td;
79         &td&td&td;?<= htmlspecialchars($j['submitted_by_name']) ?<td&td&td;
80         &td&td&td;

81         <?php if ($j['status'] === 'aprobado'): ?<td&td&td;
82             <span class="badge badge-success"&td;■ Aprobada&td;/span&td;td&td;
83         <?php else: ?<td&td&td;
84             <span class="badge badge-danger"&td;■ Rechazada&td;/span&td;td&td;
85         <?php endif: ?<td&td&td;
86     </td&td&td;
87     &td&td&td;?<= htmlspecialchars($j['reviewer_name'] ?? '-') ?<td&td&td;
88     &td&td&td;
89     <?php
90         $jData = htmlspecialchars(json_encode([
91             'student' =&td; $j['student_name'],
92             'reason' =&td; $j['reason'],
93             'notes' =&td; $j['review_notes'] ?? '',
94             'doc' =&td; $j['document_path'] ?? '',
95             'status' =&td; $j['status']
96         ]), ENT_QUOTES);
97     <td&td&td;
98     <button class="btn btn-primary" onclick="verDetalle('&td;?<= $jData ?<td;')"&td;Ver&td;/bu...
99     </td&td&td;
100     </tr&td&td;
101     <?php endforeach: ?<td&td&td;
102     </tbody&td&td;
103 </table&td&td;
104 <?php else: ?<td&td&td;
105     <p style="color: #666; text-align: center; padding: 40px;"&td;td&td;
106         No hay justificaciones revisadas aún
107     </p&td&td;
108     <?php endif: ?<td&td&td;
109 </div&td&td;
110 </div&td&td;
111
112 <!-- Modal detalle --&td;td&td;
113 <div id="modal" class="modal"&td;td&td;
114     <div class="modal-content"&td;td&td;
115         <span class="modal-close" onclick="cerrarModal()"&td;td&td;times;&td;/span&td;td&td;
116         <h3 id="modal-titulo"&td;td;Detalle de Justificación&td;/h3&td;td&td;
117         <div id="modal-body" style="margin-top: 15px;"&td;td&td;/div&td;td&td;
118     </div&td;td&td;
119 </div&td;td&td;
120
121 </script&td;td&td;
122 function verDetalle(jsonStr) {
123     const d = JSON.parse(jsonStr);
124     const estudiante = d.student;
125     const motivo = d.reason;
126     const notas = d.notes;
127     const docPath = d.doc;
128     const estado = d.status;
129
130     const badgeColor = estado === 'aprobado' ? '#d4edda' : '#f8d7da';
131     const badgeText = estado === 'aprobado' ? '#155724' : '#721c24';
132     const estadoLabel = estado === 'aprobado' ? '■ Aprobada' : '■ Rechazada';
133
134     let html = '&td;p&td;td;&td;strong&td;Estudiante:&td;/strong&td; ' + estudiante + '&td;p&td;td;';
135     html += '&td;p style="margin-top:10px;"&td;td;&td;strong&td;Estado:&td;/strong&td; &td;span style="background:...'
136     html += '&td;p style="margin-top:10px;"&td;td;&td;strong&td;Motivo:&td;/strong&td;&td;p&td;td;style="margin...'
137
138     if (notas &td;&td; notas.length &td; 0) {
139         html += '&td;p style="margin-top:10px;"&td;td;&td;strong&td;Observación del revisor:&td;/strong&td;&td;p&td;td;...'
140     }
141
142     if (docPath &td;&td; docPath.length &td; 0) {
143         html += '&td;p style="margin-top:15px;"&td;td;&td;strong&td;Documento:&td;/strong&td; &td;a href="&td;?<= BA...'
144     } else {
145         html += '&td;p style="margin-top:15px;color:#999;"&td;td;&td;em&td;Sin documento adjunto&td;/em&td;&td;p&td;td;...'
146     }
147
148     document.getElementById('modal-body').innerHTML = html;
149     document.getElementById('modal').style.display = 'block';
150 }
151
152 function cerrarModal() {
153     document.getElementById('modal').style.display = 'none';
154 }

```

```
155
156     window.onclick = function(e) {
157         const modal = document.getElementById('modal');
158         if (e.target === modal) cerrarModal();
159     }
160     </script>;

161 </body>;
162 </html>;
```

## 68. submit.php

Ruta: ArchImp/views/justifications/submit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Justificar Ausencia - 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: 700px; 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); }
15         .form-group { margin-bottom: 20px; }
16         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
17         input, textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
18         textarea { min-height: 100px; resize: vertical; }
19         button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; ...
20         button:hover { background: #218838; }
21         .info { background: #d1ecf1; color: #0c5460; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
22     </style>
23 </head>
24 <body>
25     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
26
27     <div class="container">
28         <div class="card">
29             <div class="info">
30                 ■ Complete el formulario para justificar una ausencia. Puede adjuntar documentos de respaldo (certificad...
31             </div>
32
33             <form method="POST" enctype="multipart/form-data">
34                 <div class="form-group">
35                     <label>Motivo de la Ausencia </label>
36                     <textarea name="reason" required placeholder="Describa el motivo de la ausencia...">
37                 </div>
38
39                 <div class="form-group">
40                     <label>Documento de Respaldo (opcional)</label>
41                     <input type="file" name="document" accept=".pdf,.jpg,.jpeg,.png">
42                     <small style="color: #666; display: block; margin-top: 5px;">
43                         Formatos permitidos: PDF, JPG, PNG - Máximo 5MB
44                     </small>
45                 </div>
46
47                 <button type="submit">Enviar Justificación</button>
48             </form>
49         </div>
50     </div>
51 </body>
52 </html>
```

## 69. index.php

Ruta: 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:...
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: 50...
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-w...
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             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/partial/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">&lt;?php echo $msgs[$_GET['success']] ?>&lt;/div>
107         <?php endif; ?>
108     <?php endif; ?>
109
110     <div class="page-title">
111         ■ Notificaciones
112         <span>&lt;?php echo $total ?> en total&lt;/span>
113     </div>
114
115     <div class="stats-row">
116         <div class="stat-card blue">
117             <div class="number">&lt;?php echo $unreadCount ?>&lt;/div>
118             <div class="label">No leídas&lt;/div>
119         </div>
120         <div class="stat-card">
121             <div class="number">&lt;?php echo $total ?>&lt;/div>
122             <div class="label">Total&lt;/div>
123         </div>
124     </div>
125
126     <div style="display: flex; gap: 8px; margin-bottom: 16px; flex-wrap: wrap;">
127         <?php if ($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&lt;/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&lt;/button>
135         </form>
136     </div>
137
138     <div class="notif-list">
139         <?php if (empty($notifications)): ?>
140             <div class="empty-state">
141                 <div class="icon">■&lt;/div>
142                 &lt;p>No tienes notificaciones.&lt;/p>
143             </div>
144         <?php else: ?>
145             <?php foreach ($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 &amp; $isNew) ? ' onclick="markRead(' . $n['id'] . ', this)" : ''';
154                 <?php

```

```

156 <!-- $tag --> <span class="notif-dot"></span><span class="notif-new"></span></div>
157
158 <div class="notif-icon"><img alt="Icon" /></div>
159
160
161 <div class="notif-body">
162 <div class="notif-title"><span class="notif-title"></span></div>
163 <div class="notif-message"><span class="notif-message"></span></div>
164 <div class="notif-meta">
165 <span class="notif-time"><span class="notif-time"></span></div>
166 <span class="notif-new"><span class="notif-new"></span></div>
167 <span class="notif-link-hint"><span class="notif-link-hint"></span></div>
168 </div>
169
170 <div class="notif-actions" onclick="event.stopPropagation();">
171 <div class="notif-actions">
172 <a href="#" action="notifications_mark_read" id="<id>"></a>
173 <span class="btn btn-outline btn-sm" title="Marcar como leída"></span>
174 </div>
175 <div class="notif-actions">
176 <form method="POST" action="notifications_delete" style="margin:0;">
177 <input type="hidden" name="notification_id" value="<id>"></input>
178 <input type="submit" class="btn btn-outline btn-sm" title="Eliminar"></input>
179 </div>
180 </div>
181
182 </div>
183
184 <div class="pagination">
185 <div class="pagination">
186 <div class="pagination">
187 <div class="pagination">
188 <div class="pagination">
189 <div class="pagination">
190 <div class="pagination">
191 <div class="pagination">
192 <div class="pagination">
193 <div class="pagination">
194 <div class="pagination">
195 <div class="pagination">
196 <div class="pagination">
197 <div class="pagination">
198 <div class="pagination">
199 <div class="pagination">
200 <div class="pagination">
201 <div class="pagination">
202 <div class="pagination">
203 <div class="pagination">
204 <div class="pagination">
205 <div class="pagination">
206 <div class="pagination">
207 <div class="pagination">
208 <div class="pagination">
209 <div class="pagination">
210 <div class="pagination">
211 <div class="pagination">
212 <div class="pagination">
213 <div class="pagination">
214 <div class="pagination">
215 <div class="pagination">
216 <div class="pagination">
217 <div class="pagination">
218 <div class="pagination">
219 <div class="pagination">
220 <div class="pagination">
221 <div class="pagination">
222 <div class="pagination">
223 <div class="pagination">
224 <div class="pagination">
225 <div class="pagination">
226 <div class="pagination">
227 <div class="pagination">
228 <div class="pagination">
229 <div class="pagination">
230 <div class="pagination">
231 <div class="pagination">
232 <div class="pagination">
233 <div class="pagination">

```

```
234         }
235     }).catch(function(){});
236     }, 200);
237 }
238 </script>
239
240 </body>
241 </html>
```



## 70. navbar.php

Ruta: ArchImp/views/partials/navbar.php

```
1 <?php
2 // ARCHIVO COMPLETO - Reemplaza TODO el archivo
3 // views/partials/navbar.php
4
5 $current_page = $_GET['action'] ?? 'dashboard';
6
7 // Contar notificaciones no leídas
8 $_notifCount = 0;
9 if (isset($_SESSION['user_id'])) {
10     require_once BASE_PATH . '/models/Notification.php';
11     $_notifDb = new Database();
12     $_notifModel = new Notification($_notifDb);
13     $_notifCount = $_notifModel->getUnreadCount($_SESSION['user_id']);
14 }
15 ?>
16 <style>
17 .main-navbar { background: #007bff; color: white; box-shadow: 0 2px 5px rgba(0,0,0,0.1); position: sticky; top: 0; z...
18 .navbar-container { max-width: 1400px; margin: 0 auto; padding: 0 20px; display: flex; justify-content: space-between...
19 .navbar-brand { font-size: 24px; font-weight: bold; padding: 15px 0; }
20 .navbar-menu { display: flex; gap: 20px; align-items: center; }
21 .navbar-section { position: relative; }
22 .navbar-section &gt; a { color: white; text-decoration: none; padding: 15px 12px; display: block; cursor: pointer; f...
23 .navbar-section:hover &gt; a { background: rgba(255,255,255,0.2); }
24 .dropdown { display: none; position: absolute; top: 100%; left: 0; background: white; min-width: 220px; box-shadow: ...
25 .navbar-section:hover .dropdown { display: block; }
26 .dropdown a { color: #333; text-decoration: none; padding: 12px 20px; display: block; transition: background 0.2s; b...
27 .dropdown a:hover { background: #f8f9fa; border-left-color: #007bff; }
28 .dropdown a.active { background: #e7f3ff; border-left-color: #007bff; font-weight: 600; }
29 .dropdown-divider { height: 1px; background: #e0e0e0; margin: 5px 0; }
30 .navbar-user { display: flex; align-items: center; gap: 10px; }
31 .navbar-user a { color: white; text-decoration: none; padding: 8px 15px; border-radius: 4px; transition: background ...
32 .navbar-user a:hover { background: rgba(255,255,255,0.2); }
33
34 /* Campana de notificaciones */
35 .nav-bell {
36     position: relative;
37     color: white;
38     text-decoration: none;
39     padding: 8px 12px;
40     border-radius: 4px;
41     display: inline-flex;
42     align-items: center;
43     font-size: 18px;
44     transition: background 0.3s;
45 }
46 .nav-bell:hover { background: rgba(255,255,255,0.2); }
47 .nav-bell .notif-badge {
48     position: absolute;
49     top: 4px;
50     right: 4px;
51     background: #dc3545;
52     color: #fff;
53     border-radius: 50%;
54     min-width: 17px;
55     height: 17px;
56     font-size: 0.68rem;
57     font-weight: 700;
58     display: flex;
59     align-items: center;
60     justify-content: center;
61     line-height: 1;
62     border: 2px solid #007bff;
63 }
64
65 @media (max-width: 992px) {
66     .navbar-container { flex-direction: column; align-items: flex-start; }
67     .navbar-menu { width: 100%; flex-direction: column; gap: 0; align-items: stretch; }
68     .navbar-section { width: 100%; }
69     .navbar-section &gt; a { width: 100%; }
70     .dropdown { position: static; width: 100%; box-shadow: none; background: rgba(255,255,255,0.1); }
71     .dropdown a { color: white; }
72     .dropdown a:hover { background: rgba(255,255,255,0.1); }
73 }
74 </style>
75
```

```

76 <nav class="main-navbar">
77     <div class="navbar-container">
78         <div class="navbar-brand">EcuAsist 2026</div>
79
80         <div class="navbar-menu">
81
82             <div class="navbar-section">
83                 <a href="?action=dashboard">Inicio</a>
84             </div>
85
86             <!-- ASISTENCIA -->
87             <div class="navbar-section">
88                 <a>Asistencia</a>
89                 <div class="dropdown">
90                     <div class="dropdown-item">
91                         <a href="?action=attendance_register" class="<?php if($current_page == 'attendance_register' ? 'active' : '' ?>">
92                             Registrar Asistencia
93                         </a>
94                     </div>
95
96                     <div class="dropdown-item">
97                         <a href="?action=tutor_course_attendance" class="<?php if($current_page == 'tutor_course_attendance...' ?>">
98                             Asistencia de Mi Curso
99                         </a>
100                     </div>
101
102                     <div class="dropdown-item">
103                         <a href="?action=attendance_view" class="<?php if($current_page == 'attendance_view' ? 'active' : '' ?>">
104                             Ver Asistencias
105                         </a>
106                     </div>
107                     <a href="?action=attendance_calendar" class="<?php if($current_page == 'attendance_calendar' ? 'active' : '' ?>">
108                         Calendario
109                     </a>
110                 </div>
111
112                 <div class="dropdown-item">
113                     <a href="?action=my_attendance" class="<?php if($current_page == 'my_attendance' ? 'active' : '' ?>">
114                         Mi Asistencia
115                     </a>
116                 </div>
117             </div>
118         </div>
119
120         <!-- JUSTIFICACIONES -->
121         <div class="navbar-section">
122             <a>Justificaciones</a>
123             <div class="dropdown">
124                 <div class="dropdown-item">
125                     <a href="?action=my_justifications" class="<?php if($current_page == 'my_justifications' ? 'active' : '' ?>">
126                         Mis Justificaciones
127                     </a>
128                 </div>
129                 <div class="dropdown-item">
130                     <a href="?action=pending_justifications" class="<?php if($current_page == 'pending_justifications' ? 'active' : '' ?>">
131                         Revisar Justificaciones
132                     </a>
133                 </div>
134                 <div class="dropdown-item">
135                     <a href="?action=reviewed_justifications" class="<?php if($current_page == 'reviewed_justifications' ? 'active' : '' ?>">
136                         Justificaciones Revisadas
137                     </a>
138                 </div>
139             </div>
140         </div>
141     </div>
142
143     <!-- ADMINISTRACIÓN -->
144     <div class="navbar-section">
145         <a>Administración</a>
146         <div class="dropdown">
147             <a href="?action=users" class="<?php if($current_page == 'users' ? 'active' : '' ?>">
148                 Gestión de Usuarios
149             </a>
150             <a href="?action=academic" class="<?php if($current_page == 'academic' ? 'active' : '' ?>">
151                 Configuración Académica
152             </a>
153             <a href="?action=assignments" class="<?php if($current_page == 'assignments' ? 'active' : '' ?>">
154                 Asignaciones
155         </div>
156     </div>
157 </div>

```

```

155         ■ Asignar Docente-Materia
156         </a>
157         <a href="?action=tutor_management" class="&lt;?=$current_page == 'tutor_management' ? 'active' : ...
158         ■■ Asignar Docente Tutor
159         </a>
160         <a href="?action=manage_representatives" class="&lt;?=$current_page == 'manage_representatives' ...

161         ■■■■■■ Gestión Representantes
162         </a>
163         <div class="dropdown-divider"></div>
164         <a href="?action=schedules" class="&lt;?=$current_page == 'schedules' ? 'active' : '' ?>">
165         ■ Horarios de Clases
166         </a>
167         <a href="?action=institution" class="&lt;?=$current_page == 'institution' ? 'active' : '' ?>"...
168         ■ Configuración de Institución
169         </a>
170         <div class="dropdown-divider"></div>
171         <a href="?action=backups" class="&lt;?=$current_page == 'backups' ? 'active' : '' ?>">
172         ■ RespalDOS
173         </a>
174     </div>
175 </div>
176 <?php endif; ?>
177
178 <!-- REPORTES -->
179 <?php if(Security::hasRole('autoridad')): ?>
180 <div class="navbar-section">
181     <a>■ Reportes </a>
182     <div class="dropdown">
183         <a href="?action=reports" class="&lt;?=$current_page == 'reports' ? 'active' : '' ?>">
184         ■ Generar Reportes
185         </a>
186         <a href="?action=stats" class="&lt;?=$current_page == 'stats' ? 'active' : '' ?>">
187         ■ Estadísticas
188         </a>
189     </div>
190 </div>
191 <?php endif; ?>
192
193 <!-- REPRESENTANTES -->
194 <?php if(Security::hasRole('representante')): ?>
195 <div class="navbar-section">
196     <a href="?action=my_children">■■■■■ Mis Representados</a>
197 </div>
198 <?php endif; ?>
199 </div>
200
201 <!-- Derecha: campana + usuario + logout -->
202 <div class="navbar-user">
203
204     <!-- ■ Campana de Notificaciones -->
205     <a href="?action=notifications" class="nav-bell" title="Notificaciones">
206     ■
207     <?php if ($_notifCount > 0): ?>
208         <span class="notif-badge" id="notif-badge">
209             &lt;?=$_notifCount > 99 ? '99+' : $_notifCount ?>
210         </span>
211     <?php else: ?>
212         <span class="notif-badge" id="notif-badge" style="display:none;">0</span>
213     <?php endif; ?>
214     </a>
215
216     <a href="?action=profile">
217         ■ &lt;?=$_htmlspecialchars($_SESSION['first_name']) . ' ' . $_SESSION['last_name']) ?>
218     </a>
219     <a href="?action=logout">Cerrar sesión</a>
220 </div>
221 </div>
222 </nav>
223
224 <script>
225 (function() {
226     var INTERVAL = 30000; // 30 segundos
227     var badge = document.getElementById('notif-badge');
228
229     function updateBadge(count) {
230         if (!badge) return;
231         if (count > 0) {
232             badge.textContent = count > 99 ? '99+' : count;
233             badge.style.display = 'flex';

```

```
234         } else {
235             badge.style.display = 'none';
236         }
237     }
238
239     function pollNotifications() {
240         fetch('?action=notifications_unread_json', { credentials: 'same-origin' })
241
242             .then(function(r) { return r.json(); })
243             .then(function(data) {
244                 if (typeof data.count !== 'undefined') {
245                     updateBadge(data.count);
246                 }
247             })
248             .catch(function() { /* silencioso si falla */ });
249
250         // Iniciar polling cada 30 segundos
251         setInterval(pollNotifications, INTERVAL);
252     })();
253 </script>
```

## 71. change\_password.php

Ruta: 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      <title>Cambiar Contraseña - 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: 600px; 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); }
15         .form-group { margin-bottom: 20px; }
16         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
17         input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
18         button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointe...
19         button:hover { background: #0056b3; }
20         .btn-cancel { background: #6c757d; margin-left: 10px; }
21         .btn-cancel:hover { background: #5a6268; }
22         .error { background: #f8d7da; color: #721c24; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
23     </style>
24 </head>
25 <body>
26     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
27
28     <div class="container">
29         <div class="card">
30             <?php if(isset($error)): ?>
31                 <div class="error"><?=$error ?></div>
32             <?php endif; ?>
33
34             <h2 style="margin-bottom: 20px;">Nueva Contraseña</h2>
35             <form method="POST">
36                 <div class="form-group">
37                     <label>Contraseña Actual</label>
38                     <input type="password" name="current_password" required>
39                 </div>
40
41                 <div class="form-group">
42                     <label>Nueva Contraseña</label>
43                     <input type="password" name="new_password" minlength="6" required>
44                 </div>
45
46                 <div class="form-group">
47                     <label>Confirmar Nueva Contraseña</label>
48                     <input type="password" name="confirm_password" minlength="6" required>
49                 </div>
50
51                 <button type="submit">Cambiar Contraseña</button>
52                 <button type="button" class="btn-cancel" onclick="location.href='?action=profile'">Cancelar</bu...
53             </form>
54         </div>
55     </div>
56 </body>
57 </html>
```

## 72. edit.php

Ruta: ArchImp/views/profile/edit.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Editor Perfil - 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: 600px; 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); }
15         .form-group { margin-bottom: 20px; }
16         label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
17         input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
18         button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; ...
19         button:hover { background: #218838; }
20         .btn-cancel { background: #6c757d; margin-left: 10px; }
21         .btn-cancel:hover { background: #5a6268; }
22     </style>
23 </head>
24 <body>
25     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
26
27     <div class="container">
28         <div class="card">
29             <h2 style="margin-bottom: 20px;">Actualizar Información</h2>
30             <form method="POST">
31                 <div class="form-group">
32                     <label>Nombres</label>
33                     <input type="text" name="first_name" value="<?= $user['first_name'] ?>" required>
34                 </div>
35
36                 <div class="form-group">
37                     <label>Apellidos</label>
38                     <input type="text" name="last_name" value="<?= $user['last_name'] ?>" required>
39                 </div>
40
41                 <div class="form-group">
42                     <label>Email</label>
43                     <input type="email" name="email" value="<?= $user['email'] ?>" required>
44                 </div>
45
46                 <div class="form-group">
47                     <label>Teléfono</label>
48                     <input type="text" name="phone" value="<?= $user['phone'] ?>">
49                 </div>
50
51                 <button type="submit">Guardar Cambios</button>
52                 <button type="button" class="btn-cancel" onclick="location.href='?action=profile'">Cancelar</bu...
53             </form>
54         </div>
55     </div>
56 </body>
57 </html>
```

## 73. view.php

Ruta: ArchImp/views/profile/view.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Mi Perfil - 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: 800px; 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-bot...
15        .profile-header { display: flex; align-items: center; gap: 20px; margin-bottom: 30px; }
16        .avatar { width: 100px; height: 100px; border-radius: 50%; background: linear-gradient(135deg, #667eea 0%, #764b...
17        .info-row { display: flex; padding: 15px 0; border-bottom: 1px solid #f0f0f0; }
18        .info-row:last-child { border-bottom: none; }
19        .label { font-weight: bold; color: #666; width: 150px; }
20        .value { color: #333; flex: 1; }
21        .btn { display: inline-block; padding: 10px 20px; background: #007bff; color: white; text-decoration: none; bord...
22        .btn:hover { background: #0056b3; }
23        .btn-secondary { background: #6c757d; }
24        .btn-secondary:hover { background: #5a6268; }
25        .badge { display: inline-block; padding: 5px 10px; border-radius: 3px; background: #28a745; color: white; font-s...
26        .success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
27    </style>
28 </head>
29 <body>
30     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
31
32     <div class="container">
33         <?php if(isset($_GET['success'])): ?>
34             <div class="success">✓ Perfil actualizado correctamente</div>
35         <?php endif; ?>
36
37         <?php if(isset($_GET['password_changed'])): ?>
38             <div class="success">✓ Contraseña cambiada correctamente</div>
39         <?php endif; ?>
40
41         <div class="card">
42             <div class="profile-header">
43                 <div class="avatar">
44                     <?php $avatar = strtoupper(substr($user['first_name'], 0, 1) . substr($user['last_name'], 0, 1)); ?>
45                 </div>
46                 <div>
47                     <h2><?php $name = $user['first_name'] . ' ' . $user['last_name']; ?></h2>
48                     <p style="color: #666;">@<?php $username = $user['username']; ?></p>
49                 </div>
50             </div>
51
52             <div class="info-row">
53                 <span class="label">Email:</span>
54                 <span class="value"><?php $email = $user['email']; ?></span>
55             </div>
56
57             <div class="info-row">
58                 <span class="label">Cédula:</span>
59                 <span class="value"><?php $dni = $user['dni']; ?> 'No registrada' ?></span>
60             </div>
61
62             <div class="info-row">
63                 <span class="label">Teléfono:</span>
64                 <span class="value"><?php $phone = $user['phone']; ?> 'No registrado' ?></span>
65             </div>
66
67             <div class="info-row">
68                 <span class="label">Roles:</span>
69                 <span class="value">
70                     <?php foreach($roles as $role): ?>
71                         <span class="badge"><?php $role = ucfirst($role); ?></span>
72                     <?php endforeach; ?>
73                 </span>
74             </div>
75         </div>
76     </div>
```

```
76         <div class="info-row">
77             <span class="label">Miembro desde:</span>
78             <span class="value"><?= date('d/m/Y', strtotime($user['created_at'])) ?></span>
79         </div>
80
81         <div style="margin-top: 20px;">
82             <a href="?action=edit_profile" class="btn">Editar Perfil</a>
83             <a href="?action=change_password" class="btn btn-secondary">Cambiar Contraseña</a>
84         </div>
85     </div>
86 </div>
87 </body>
88 </html>
```





```

76         <input type="date" name="end_date" value="&lt;?= $endDate ?? date('Y-m-d') ?&gt;" required&gt;
77     </div&gt;
78
79     <div class="btn-group"&gt;
80         <button type="submit" name="preview" class="btn-preview"&gt;
81
82             ■■ Vista Previa
83         </button&gt;
84         <button type="button" class="btn-pdf" onclick="generateReport('pdf')" &lt;?= empty($data) ? 'disa...
85             ■ Generar PDF
86         </button&gt;
87         <button type="button" class="btn-excel" onclick="generateReport('excel')" &lt;?= empty($data) ? '...
88             ■ Generar Excel
89         </button&gt;
90     </div&gt;
91 </div&gt;
92
93 <?php if(!empty($data)): ?&gt;
94 <div class="card"&gt;
95     <h2&gt;Vista Previa del Reporte</h2&gt;
96
97     <div class="info"&gt;
98         <strong&gt;Curso:</strong&gt; &lt;?= $course['name'] ?&gt; - &lt;?= $course['shift_name'] ?&gt;&lt;...
99         <strong&gt;Período:</strong&gt; &lt;?= date('d/m/Y', strtotime($startDate)) ?&gt; al &lt;?= date('...
100        <strong&gt;Total Registros:</strong&gt; &lt;?= count($data) ?&gt;
101    </div&gt;
102
103    <?php
104    $totalPresente = count(array_filter($data, fn($d) => $d['status'] === 'presente'));
105    $totalAusente = count(array_filter($data, fn($d) => $d['status'] === 'ausente'));
106    $totalTardanza = count(array_filter($data, fn($d) => $d['status'] === 'tardanza'));
107    $totalJustificado = count(array_filter($data, fn($d) => $d['status'] === 'justificado'));
108    $porcentaje = count($data) &gt; 0 ? round(($totalPresente / count($data)) * 100, 1) : 0;
109    ?&gt;
110
111    <div class="stats"&gt;
112        <div class="stat-item"&gt;
113            <div class="stat-number" style="color: #28a745;"&gt;&lt;?= $totalPresente ?&gt;</div&gt;
114            <div class="stat-label"&gt;Presentes</div&gt;
115        </div&gt;
116        <div class="stat-item"&gt;
117            <div class="stat-number" style="color: #dc3545;"&gt;&lt;?= $totalAusente ?&gt;</div&gt;
118            <div class="stat-label"&gt;Ausentes</div&gt;
119        </div&gt;
120        <div class="stat-item"&gt;
121            <div class="stat-number" style="color: #ffc107;"&gt;&lt;?= $totalTardanza ?&gt;</div&gt;
122            <div class="stat-label"&gt;Tardanzas</div&gt;
123        </div&gt;
124        <div class="stat-item"&gt;
125            <div class="stat-number" style="color: #17a2b8;"&gt;&lt;?= $totalJustificado ?&gt;</div&gt;
126            <div class="stat-label"&gt;Justificados</div&gt;
127        </div&gt;
128        <div class="stat-item"&gt;
129            <div class="stat-number" style="color: #6f42c1;"&gt;&lt;?= $porcentaje ?&gt;%</div&gt;
130            <div class="stat-label"&gt;Asistencia</div&gt;
131        </div&gt;
132    </div&gt;
133
134    <table&gt;
135        <thead&gt;
136            <tr&gt;
137                <th&gt;Fecha</th&gt;
138                <th&gt;Estudiante</th&gt;
139                <th&gt;Asignatura</th&gt;
140                <th&gt;Hora</th&gt;
141                <th&gt;Estado</th&gt;
142                <th&gt;Observación</th&gt;
143            </tr&gt;
144        </thead&gt;
145        <tbody&gt;
146            <?php foreach($data as $row): ?&gt;
147                <tr&gt;
148                    <td&gt;&lt;?= date('d/m/Y', strtotime($row['date'])) ?&gt;</td&gt;
149                    <td&gt;&lt;?= $row['student_name'] ?&gt;</td&gt;
150                    <td&gt;&lt;?= $row['subject_name'] ?&gt;</td&gt;
151                    <td&gt;&lt;?= $row['hour_period'] ?&gt;</td&gt;
152                    <td&gt;
153                        <span class="badge badge-&lt;?= $row['status'] ?&gt;"&gt;
154                            &lt;?= ucfirst($row['status']) ?&gt;

```

```

155         </span>;
156         </td>;
157         <td><?php $row['observation'] ? : '-' ?></td>;
158     </tr>;
159     <?php endforeach; ?>
160 </tbody>;
161 </table>;
162 </div>;
163 <?php elseif($_SERVER['REQUEST_METHOD'] === 'POST') : ?>
164 <div class="card">
165     <p style="text-align: center; color: #666; padding: 40px;">
166         No hay registros de asistencia para el período seleccionado
167     </p>
168 </div>;
169 <?php endif; ?>
170 </div>;
171
172 <script>
173     function generateReport(type) {
174         const form = document.getElementById('report-form');
175
176         if (!form.checkValidity()) {
177             form.reportValidity();
178             return;
179         }
180
181         // Crear un formulario temporal
182         const tempForm = document.createElement('form');
183         tempForm.method = 'POST';
184         tempForm.style.display = 'none';
185
186         if (type === 'pdf') {
187             tempForm.action = '?action=generate_pdf';
188         } else {
189             tempForm.action = '?action=generate_excel';
190         }
191
192         // Copiar valores del formulario original
193         const courseId = form.querySelector('[name="course_id"]').value;
194         const startDate = form.querySelector('[name="start_date"]').value;
195         const endDate = form.querySelector('[name="end_date"]').value;
196
197         // Crear inputs ocultos
198         const inputs = [
199             { name: 'course_id', value: courseId },
200             { name: 'start_date', value: startDate },
201             { name: 'end_date', value: endDate }
202         ];
203
204         inputs.forEach(input => {
205             const hiddenInput = document.createElement('input');
206             hiddenInput.type = 'hidden';
207             hiddenInput.name = input.name;
208             hiddenInput.value = input.value;
209             tempForm.appendChild(hiddenInput);
210         });
211
212         // Agregar al body, enviar y eliminar
213         document.body.appendChild(tempForm);
214         tempForm.submit();
215         document.body.removeChild(tempForm);
216     }
217 </script>;
218 </body>;
219 </html>;

```

## 75. child\_attendance.php

Ruta: 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-bot...
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: 2...
24         .stat-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 20px; border-...
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 = $total &gt; 0 ? round(($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 $att): ?>
74                         <tr>
75                             <td><?php echo date('d/m/Y', strtotime($att['date'])); ?></td>
```

```
76         &lt;td&gt;&lt;?= $att['subject_name'] ?&gt;&lt;/td&gt;
77         &lt;td&gt;&lt;?= $att['hour_period'] ?&gt;&lt;/td&gt;
78         &lt;td&gt;
79             &lt;span class="badge badge-&lt;?= $att['status'] ?&gt;"&gt;
80                 &lt;?= ucfirst($att['status']) ?&gt;
81             &lt;/span&gt;
82         &lt;/td&gt;
83         &lt;td&gt;&lt;?= $att['observation'] ? : '-' ?&gt;&lt;/td&gt;
84     &lt;/tr&gt;
85     &lt;?php endforeach; ?&gt;
86 &lt;/tbody&gt;
87 &lt;/table&gt;
88 &lt;/div&gt;
89 &lt;/div&gt;
90 &lt;/body&gt;
91 &lt;/html&gt;
```

## 76. manage.php

Ruta: 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      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9      <style>
10         body { padding-top: 70px; background: #f4f4f4; }
11     </style>
12 </head>
13 <body>
14     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
15
16     <div class="container mt-4">
17
18         <?php if(isset($_GET['success'])): ?>
19             <div class="alert alert-success alert-dismissible fade show">
20                 ✓ Relación asignada correctamente
21                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
22             </div>
23         <?php endif; ?>
24
25         <?php if(isset($_GET['removed'])): ?>
26             <div class="alert alert-success alert-dismissible fade show">
27                 ✓ Relación eliminada correctamente
28                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
29             </div>
30         <?php endif; ?>
31
32         <?php if(isset($_GET['error'])): ?>
33             <div class="alert alert-danger alert-dismissible fade show">
34                 ✗ Error al procesar la solicitud
35                 <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
36             </div>
37         <?php endif; ?>
38
39         <!-- Formulario de asignación -->
40         <div class="card mb-4">
41             <div class="card-header bg-primary text-white">
42                 <h5 class="mb-0">■ Asignar Estudiante a Representante</h5>
43             </div>
44             <div class="card-body">
45                 <form method="POST">
46                     <div class="row">
47                         <div class="col-md-6 mb-3">
48                             <label class="form-label">Representante *</label>
49                             <select name="representative_id" class="form-select" required>
50                                 <option value="">Seleccionar representante...</option>
51                                 <?php foreach($representatives as $rep): ?>
52                                     <option value="<?=$rep['id']"><?>";>
53                                     <?=$rep['last_name'] . ' ' . $rep['first_name']">
54                                     (<?=$rep['dni'] ?? 'Sin cédula' ">);
55                                 </option>
56                             <?php endforeach; ?>
57                         </select>
58                     </div>
59
60                     <div class="col-md-6 mb-3">
61                         <label class="form-label">Estudiante *</label>
62                         <select name="student_id" class="form-select" required>
63                             <option value="">Seleccionar estudiante...</option>
64                             <?php foreach($students as $student): ?>
65                                 <option value="<?=$student['id']"><?>";>
66                                 <?=$student['last_name'] . ' ' . $student['first_name']">
67                                 </option>
68                             <?php endforeach; ?>
69                         </select>
70                     </div>
71
72                     <div class="col-md-6 mb-3">
73                         <label class="form-label">Parentesco *</label>
74                         <select name="relationship" class="form-select" required>
75                             <option value="">Seleccionar parentesco...</option>
```

```

76         <option value="Padre">Padre</option>
77         <option value="Madre">Madre</option>
78         <option value="Tutor Legal">Tutor Legal</option>
79         <option value="Abuelo/a">Abuelo/a</option>
80         <option value="Tío/a">Tío/a</option>

81         <option value="Hermano/a">Hermano/a</option>
82         <option value="Otro">Otro</option>
83     </select>
84 </div>
85
86     <div class="col-md-6 mb-3">
87         <label class="form-label">&nbsp;</label>
88         <div class="form-check" style="padding-top: 8px;">
89             <input type="checkbox" name="is_primary" value="1" class="form-check-input" id="isPri...
90             <label class="form-check-label" for="isPrimary">
91                 <strong>Representante Principal</strong>
92             </label>
93         </div>
94     </div>
95 </div>
96
97     <button type="submit" class="btn btn-success">
98         ■ Asignar Relación
99     </button>
100 </form>
101 </div>
102 </div>
103
104 <!-- Filtros -->
105 <div class="card mb-4">
106     <div class="card-header bg-light">
107         <h5 class="mb-0">■ Filtros de Búsqueda</h5>
108     </div>
109     <div class="card-body">
110         <div class="row">
111             <div class="col-md-4 mb-3">
112                 <label class="form-label">Buscar por Representante</label>
113                 <input type="text" id="filterRepresentative" class="form-control" placeholder="Nombre del rep...
114             </div>
115             <div class="col-md-4 mb-3">
116                 <label class="form-label">Buscar por Estudiante</label>
117                 <input type="text" id="filterStudent" class="form-control" placeholder="Nombre del estudiante...
118             </div>
119             <div class="col-md-4 mb-3">
120                 <label class="form-label">Buscar por Curso</label>
121                 <input type="text" id="filterCourse" class="form-control" placeholder="Nombre del curso...">
122             </div>
123         </div>
124         <button onclick="clearFilters()" class="btn btn-secondary btn-sm">
125             ■■ Limpiar Filtros
126         </button>
127     </div>
128 </div>
129
130 <!-- Lista de relaciones -->
131 <div class="card">
132     <div class="card-header bg-light d-flex justify-content-between align-items-center">
133         <h5 class="mb-0">■ Representantes y sus Estudiantes</h5>
134         <span class="badge bg-primary" id="totalCount">
135             <?php
136                 $totalRelations = 0;
137                 foreach($representatives as $rep) {
138                     $totalRelations += count($this->representativeModel->getStudentsByRepresentative($rep['id']...
139                 }
140                 echo $totalRelations;
141                 <?> relaciones
142             </span>
143     </div>
144     <div class="card-body">
145         <?php
146             $hasRelations = false;
147             foreach($representatives as $rep):
148                 $children = $this->representativeModel->getStudentsByRepresentative($rep['id']);
149                 if(count($children) > 0):
150                     $hasRelations = true;
151                 <?>
152             </div>
153             <div class="representative-block mb-4 p-3 border rounded"
154                 data-representative="<?> strtolower($rep['last_name']) . ' ' . $rep['first_name']) <?>";

```

```

155         <div class="d-flex justify-content-between align-items-center mb-2">
156             <h5 class="mb-0">
157                 ■ <?php echo $rep['last_name'] . ' ' . $rep['first_name'] ?>
158             </h5>
159             <small class="text-muted"><?php echo $rep['email'] ?></small>
160         </div>
161
162         <div class="table-responsive">
163             <table class="table table-sm table-hover mb-0">
164                 <thead class="table-light">
165                     <tr>
166                         <th>Estudiante</th>
167                         <th>Parentesco</th>
168                         <th>Curso</th>
169                         <th>Tipo</th>
170                         <th>Acciones</th>
171                     </tr>
172                 </thead>
173                 <tbody>
174                     <?php foreach($children as $child): ?>
175                         <tr class="student-row">
176                             data-student="<?php echo strtolower($child['last_name'] . ' ' . $child['first_name']) ?>..."
177                             data-course="<?php echo strtolower($child['course_name']) ?>..."
178                             <td>
179                                 <strong><?php echo $child['last_name'] . ' ' . $child['first_name'] ?></strong>
180                             </td>
181                             <td><?php echo $child['relationship'] ?></td>
182                             <td>
183                                 <?php if($child['course_name']): ?>
184                                     <span class="badge bg-info"><?php echo $child['course_name'] ?></span>
185                                 <?php else: ?>
186                                     <span class="text-muted">Sin curso</span>
187                                 <?php endif; ?>
188                             </td>
189                             <td>
190                                 <?php if($child['is_primary']): ?>
191                                     <span class="badge bg-warning text-dark">■ Principal</span>
192                                 <?php else: ?>
193                                     <span class="badge bg-secondary">Secundario</span>
194                                 <?php endif; ?>
195                             </td>
196                             <td>
197                                 <button onclick="confirmRemove(<?php echo $rep['id'] ?>,<?php echo $child['id'] ?>..."
198                                     class="btn btn-danger btn-sm">
199                                     ✕ Eliminar
200                                 </button>
201                             </td>
202                         </tr>
203                     <?php endforeach; ?>
204                 </tbody>
205             </table>
206         </div>
207     </div>
208     <?php
209     endif;
210 endforeach;
211
212     if(!$hasRelations):
213         <div class="alert alert-secondary text-center">
214             ■ No hay relaciones representante-estudiante registradas.
215             <br>Use el formulario superior para crear una.
216         </div>
217     <?php endif; ?>
218 </div>
219 </div>
220 </div>
221 </div>
222
223 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
224 </script>
225 // Filtros en tiempo real
226 const filterRepresentative = document.getElementById('filterRepresentative');
227 const filterStudent = document.getElementById('filterStudent');
228 const filterCourse = document.getElementById('filterCourse');
229
230 function applyFilters() {
231     const repValue = filterRepresentative.value.toLowerCase();
232     const stuValue = filterStudent.value.toLowerCase();
233     const courValue = filterCourse.value.toLowerCase();

```



```

234
235     const blocks = document.querySelectorAll('.representative-block');
236     let visibleCount = 0;
237
238     blocks.forEach(block => {
239         const repName = block.dataset.representative;
240         const rows = block.querySelectorAll('.student-row');
241
242         let hasVisibleRows = false;
243
244         rows.forEach(row => {
245             const stuName = row.dataset.student;
246             const course = row.dataset.course;
247
248             const matchRep = !repValue || repName.includes(repValue);
249             const matchStu = !stuValue || stuName.includes(stuValue);
250             const matchCou = !courValue || course.includes(courValue);
251
252             if (matchRep && matchStu && matchCou) {
253                 row.style.display = '';
254                 hasVisibleRows = true;
255                 visibleCount++;
256             } else {
257                 row.style.display = 'none';
258             }
259         });
260
261         block.style.display = hasVisibleRows ? '' : 'none';
262     });
263
264     // Actualizar contador
265     document.getElementById('totalCount').textContent = visibleCount + ' relaciones';
266 }
267
268 function clearFilters() {
269     filterRepresentative.value = '';
270     filterStudent.value = '';
271     filterCourse.value = '';
272     applyFilters();
273 }
274
275 filterRepresentative.addEventListener('input', applyFilters);
276 filterStudent.addEventListener('input', applyFilters);
277 filterCourse.addEventListener('input', applyFilters);
278
279 // Modal de confirmación
280 function confirmRemove(repId, studentId, repName, studentName) {
281     const modal = document.createElement('div');
282     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0...';
283
284     const modalContent = document.createElement('div');
285     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-sh...';
286
287     modalContent.innerHTML = `
288         <h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Eliminar Relación</h3>
289         <p style="margin: 0 0 20px 0; color: #666;">
290             ¿Está seguro de eliminar la relación entre:
291             <p style="margin: 0 0 10px 0; padding: 10px; background: #f8f9fa; border-radius: 4px;">
292                 <strong>Representante:</strong> ${repName}<br>
293                 <strong>Estudiante:</strong> ${studentName}
294             </p>
295             <p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #f8d7da; padding: 10px; borde...
296                 <strong>Advertencia:</strong> El representante ya no podrá ver la información del est...
297             </p>
298             <div style="display: flex; gap: 10px; justify-content: flex-end;">
299                 <button type="button" id="cancelRemoveBtn" style="padding: 10px 20px; background: #6c757d; color:...
300                     Cancelar
301                 </button>
302                 <button type="button" id="confirmRemoveBtn" style="padding: 10px 20px; background: #dc3545; color...
303                     Sí, Eliminar
304                 </button>
305             </div>
306         `;
307
308     modal.appendChild(modalContent);
309     document.body.appendChild(modal);
310
311     document.getElementById('confirmRemoveBtn').onclick = function() {
312         document.body.removeChild(modal);

```

```
313         window.location.href = '?action=remove_representative&rep_id=' + repId + '&student_id=' + studen...
314     };
315
316     document.getElementById('cancelRemoveBtn').onclick = function() {
317         document.body.removeChild(modal);
318     };
319 }
320 </script>
321 </body>
322 </html>
```

## 77. my\_children.php

Ruta: 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     <title>Mis Representados - 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        .student-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 20px; }
15        .student-card { background: white; padding: 25px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
16        .student-card h3 { color: #007bff; margin-bottom: 15px; }
17        .info-row { padding: 8px 0; border-bottom: 1px solid #f0f0f0; }
18        .info-row:last-child { border-bottom: none; }
19        .label { font-weight: bold; color: #666; display: inline-block; width: 120px; }
20        .btn { display: inline-block; padding: 10px 20px; background: #007bff; color: white; text-decoration: none; bord...
21        .btn:hover { background: #0056b3; }
22    </style>
23 </head>
24 <body>
25     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
26
27     <div class="container">
28         <?php if(count($children) > 0): ?>
29             <div class="student-grid">
30                 <?php foreach($children as $child): ?>
31                     <div class="student-card">
32                         <h3><?=$child['last_name'] . ' ' . $child['first_name'] ?></h3>
33
34                         <div class="info-row">
35                             <span class="label">Cédula:</span>
36                             <?=$child['dni'] ?? 'No registrada' ?>
37                         </div>
38
39                         <div class="info-row">
40                             <span class="label">Parentesco:</span>
41                             <?=$child['relationship'] ?>
42                             <?php if($child['is_primary']): ?>
43                                 <strong style="color: #ffc107;">(Principal)</strong>
44                             <?php endif; ?>
45                         </div>
46
47                         <div class="info-row">
48                             <span class="label">Curso:</span>
49                             <?=$child['course_name'] ?? 'Sin asignar' ?>
50                         </div>
51
52                         <div class="info-row">
53                             <span class="label">Jornada:</span>
54                             <?=$child['shift_name'] ? ucfirst($child['shift_name']) : '-' ?>
55                         </div>
56
57                         <a href="?action=child_attendance&student_id=<?=$child['id'] ?>" class="btn">
58                             Ver Asistencia
59                         </a>
60                     </div>
61                 <?php endforeach; ?>
62             </div>
63             <?php else: ?>
64                 <div class="student-card" style="text-align: center; padding: 40px;">
65                     <p style="color: #666;">No tiene estudiantes asignados como representante.</p>
66                 </div>
67             <?php endif; ?>
68         </div>
69     </body>
70 </html>
```

## 78. index.php

Ruta: ArchImp/views/schedule/index.php

```
1  <?php Security::requireLogin(); ?>
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Gestión de Horarios - 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); }
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         .btn { padding: 8px 15px; background: #007bff; color: white; border: none; border-radius: 4px; text-decoration: ...
16         .btn:hover { background: #0056b3; }
17         h2 { margin-bottom: 20px; color: #333; }
18     </style>
19 </head>
20 <body>
21     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
22
23     <div class="container">
24         <div class="card">
25             <h2>Gestión de Horarios por Curso</h2>
26             <p style="color: #666; margin-bottom: 20px;">
27                 Seleccione un curso para configurar su horario de clases
28             </p>
29             <table>
30                 <thead>
31                     <tr>
32                         <th>Curso</th>
33                         <th>Nivel</th>
34                         <th>Paralelo</th>
35                         <th>Jornada</th>
36                         <th>Acción</th>
37                     </tr>
38                 </thead>
39                 <tbody>
40                     <?php foreach($courses as $course): ?>
41                         <tr>
42                             <td><?=$course['name'] ?></td>
43                             <td><?=$course['grade_level'] ?></td>
44                             <td><?=$course['parallel'] ?></td>
45                             <td><?=$course['shift_name'] ?></td>
46                             <td>
47                                 <a href="?action=manage_schedule&course_id=<?=$course['id'] ?>" class="btn">...
48                                     Configurar Horario
49                                 </a>
50                             </td>
51                         </tr>
52                     </tbody>
53                 </table>
54             </div>
55         </div>
56     </div>
57 </body>
58 </html>
```

## 79. manage.php

Ruta: ArchImp/views/schedule/manage.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Horario de <? $course['name'] ?> - EcuAsist</title>
7     <style>
8         * { margin: 0; padding: 0; box-sizing: border-box; }
9         body { font-family: Arial, sans-serif; background: #f4f4f4; }
10        .container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
11        .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot...
12        .form-group { margin-bottom: 15px; }
13        .error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
14        label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
15        select, input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
16        button { padding: 10px 20px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointe...
17        button:hover { background: #218838; }
18        .btn-danger { background: #dc3545; padding: 5px 10px; font-size: 12px; }
19        .btn-danger:hover { background: #c82323; }
20        table { width: 100%; border-collapse: collapse; margin-top: 20px; }
21        th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
22        th { background: #f8f9fa; font-weight: bold; }
23        .success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
24        .grid { display: grid; grid-template-columns: 1fr 1fr; gap: 20px; }
25        h2 { margin-bottom: 20px; color: #333; }
26        .course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
27    </style>
28 </head>
29 <body>
30     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
31
32     <div class="container">
33         <?php if(isset($_GET['success'])): ?>
34             <div class="success">✓ Clase agregada al horario</div>
35         <?php endif; ?>
36         <?php if(isset($_GET['deleted'])): ?>
37             <div class="success">✓ Clase eliminada del horario</div>
38         <?php endif; ?>
39         <?php if(isset($_GET['error'])): ?>
40             <div class="error">✗ <? htmlspecialchars($_GET['error']) ?></div>
41         <?php endif; ?>
42
43         <div class="course-info">
44             <h2><? $course['name'] ?></h2>
45             <p><strong>Nivel:</strong> <? $course['grade_level'] ?> | <strong>Paralelo:</strong> <? $course['grade_level'] ?></p>
46         </div>
47
48         <div class="grid">
49             <div class="card">
50                 <h2>Agregar Clase al Horario</h2>
51                 <form method="POST" id="scheduleForm">
52                     <div class="form-group">
53                         <label>Día de la Semana</label>
54                         <select name="day_of_week" id="day_of_week" required>
55                             <option value="">Seleccionar...</option>
56                             <option value="lunes">Lunes</option>
57                             <option value="martes">Martes</option>
58                             <option value="miercoles">Miércoles</option>
59                             <option value="jueves">Jueves</option>
60                             <option value="viernes">Viernes</option>
61                             <option value="sabado">Sábado</option>
62                         </select>
63                     </div>
64
65                     <div class="form-group">
66                         <label>Número de Hora</label>
67                         <select name="period_number" id="period_number" required>
68                             <option value="">Seleccionar...</option>
69                         <?php
70                             $maxHours = strpos($course['grade_level'], 'Técnico') !== false ? 8 : 7;
71                             for($i = 1; $i <= $maxHours; $i++):
72                                 <option value="<? $i ?>"><? $i ?> hora</option>
73                             <?php endfor; ?>
74                         </select>
75                     </div>
76                 </form>
77             </div>
78         </div>
79     </div>
80 </body>
81 </html>
```

```

76         </div>;
77
78         <div class="form-group">
79             <label>Asignatura</label>
80             <select name="subject_id" id="subject_id" required>
81
82                 <option value="">Seleccionar...</option>
83             </select>
84         </div>
85
86         <div class="form-group">
87             <label>Docente Asignado</label>
88             <input type="text" id="teacher_name" readonly style="background: #f0f0f0;" placeholder="Se as...
89             <input type="hidden" name="teacher_id" id="teacher_id">
90         </div>
91
92         <button type="submit" id="submitBtn" disabled style="background: #6c757d;">Agregar Clase</b...
93     </form>;
94
95     <script>
96     const courseId = <?=> $course['id'] ?>;
97
98     // Cargar asignaturas del curso al cargar la página
99     fetch('?action=get_course_subjects_schedule&course_id=' + courseId)
100     .then(response => response.json())
101     .then(data => {
102         const subjectSelect = document.getElementById('subject_id');
103         if (data.length === 0) {
104             subjectSelect.innerHTML = '<option value="">No hay asignaturas asignadas a este curso&...
105         } else {
106             subjectSelect.innerHTML = '<option value="">Seleccionar...</option>';
107             data.forEach(item => {
108                 const option = document.createElement('option');
109                 option.value = item.subject_id;
110                 option.textContent = item.subject_name;
111                 option.setAttribute('data-teacher-id', item.teacher_id);
112                 option.setAttribute('data-teacher-name', item.teacher_name);
113                 subjectSelect.appendChild(option);
114             });
115         }
116     });
117
118     // Al seleccionar asignatura, mostrar docente automáticamente
119     document.getElementById('subject_id').addEventListener('change', function() {
120         const selectedOption = this.options[this.selectedIndex];
121         const teacherId = selectedOption.getAttribute('data-teacher-id');
122         const teacherName = selectedOption.getAttribute('data-teacher-name');
123
124         if (teacherId && teacherName) {
125             document.getElementById('teacher_id').value = teacherId;
126             document.getElementById('teacher_name').value = teacherName;
127             checkFormComplete();
128         } else {
129             document.getElementById('teacher_id').value = '';
130             document.getElementById('teacher_name').value = '';
131             document.getElementById('submitBtn').disabled = true;
132             document.getElementById('submitBtn').style.background = '#6c757d';
133         }
134     });
135
136     document.getElementById('day_of_week').addEventListener('change', checkFormComplete);
137     document.getElementById('period_number').addEventListener('change', checkFormComplete);
138
139     function checkFormComplete() {
140         const day = document.getElementById('day_of_week').value;
141         const period = document.getElementById('period_number').value;
142         const subject = document.getElementById('subject_id').value;
143         const teacher = document.getElementById('teacher_id').value;
144
145         const btn = document.getElementById('submitBtn');
146         if (day && period && subject && teacher) {
147             btn.disabled = false;
148             btn.style.background = '#28a745';
149         } else {
150             btn.disabled = true;
151             btn.style.background = '#6c757d';
152         }
153     }
154     document.getElementById('day_of_week').addEventListener('change', checkExistingSchedule);

```

```

155     document.getElementById('period_number').addEventListener('change', checkExistingSchedule);
156
157     function checkExistingSchedule() {
158         const day = document.getElementById('day_of_week').value;
159         const period = document.getElementById('period_number').value;
160
161         if (!day || !period) return;
162
163         fetch('?action=check_schedule_conflict&course_id=' + courseId + '&day=' + day + '&period...'
164             .then(response => response.json())
165             .then(data => {
166                 if (data.exists) {
167                     showWarning('■ ■ Esta hora ya está ocupada con ${data.subject_name} (${data.teacher_name})...
168                     document.getElementById('submitBtn').disabled = true;
169                     document.getElementById('submitBtn').style.background = '#6c757d';
170                 }
171             }));
172     }
173
174     function showWarning(message) {
175         // Remover advertencia anterior si existe
176         const existingWarning = document.getElementById('scheduleWarning');
177         if (existingWarning) {
178             existingWarning.remove();
179         }
180
181         const warning = document.createElement('div');
182         warning.id = 'scheduleWarning';
183         warning.style.cssText = 'background: #fff3cd; color: #856404; padding: 12px; border-radius: 4px; mar...
184         warning.innerHTML = message;
185
186         document.getElementById('scheduleForm').appendChild(warning);
187
188         setTimeout(() => {
189             warning.style.transition = 'opacity 0.5s';
190             warning.style.opacity = '0';
191             setTimeout(() => warning.remove(), 500);
192         }, 4000);
193     }
194     </script>
195 </div>
196
197 <div class="card">
198     <h2>Horario Actual</h2>
199     <table>
200         <thead>
201             <tr>
202                 <th>Día</th>
203                 <th>Hora</th>
204                 <th>Asignatura</th>
205                 <th>Docente</th>
206                 <th>Acción</th>
207             </tr>
208         </thead>
209         <tbody>
210             <?php if(empty($schedule)): ?>
211                 <tr>
212                     <td colspan="5" style="text-align:center">>No hay clases en el horario</td>
213                 </tr>
214             <?php else: ?>
215                 <?php foreach($schedule as $class): ?>
216                     <tr>
217                         <td><?php ucfirst($class['day_of_week']) ?></td>
218                         <td><?php $class['period_number'] ?></td>
219                         <td><?php $class['subject_name'] ?></td>
220                         <td><?php $class['teacher_name'] ?></td>
221                         <td>
222                             <form method="POST" action="?action=delete_schedule_class" style="display: inline...
223                             <input type="hidden" name="schedule_id" value="<?php $class['id'] ?>">
224                             <input type="hidden" name="course_id" value="<?php $course['id'] ?>">
225                             <button type="submit" class="btn-danger" onclick="return confirm('¿Eliminar e...
226                                 Eliminar
227                             </button>
228                         </form>
229                     </td>
230                 </tr>
231             <?php endforeach; ?>
232             <?php endif; ?>
233         </tbody>

```

```
234         </table>;
235     </div>;
236 </div>;
237 </div>;
238 </body>;
239 </html>;
```



## 80. index.php

Ruta: ArchImp/views/stats/index.php

```
1 <?php Security::requireLogin(); ?>
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <title>Estadísticas - 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        .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bot...
15        .stats-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px; margin-bott...
16        .stat-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 25px; border-...
17        .stat-number { font-size: 36px; font-weight: bold; margin-bottom: 5px; }
18        .stat-label { font-size: 14px; opacity: 0.9; }
19        table { width: 100%; border-collapse: collapse; margin-top: 15px; }
20        th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
21        th { background: #f8f9fa; font-weight: bold; }
22        .progress-bar { background: #e9ecef; border-radius: 4px; height: 20px; overflow: hidden; }
23        .progress-fill { background: #28a745; height: 100%; transition: width 0.3s; }
24        .form-group { margin-bottom: 15px; display: inline-block; margin-right: 15px; }
25        label { display: block; margin-bottom: 5px; font-weight: bold; }
26        input, button { padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
27        button { background: #007bff; color: white; border: none; cursor: pointer; }
28        button:hover { background: #0056b3; }
29        h2 { margin-bottom: 20px; color: #333; }
30    </style>
31 </head>
32 <body>
33     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
34
35     <div class="container">
36         <div class="card">
37             <form method="GET">
38                 <input type="hidden" name="action" value="stats">
39                 <div class="form-group">
40                     <label>Fecha Inicio</label>
41                     <input type="date" name="start_date" value="<?=$_GET['start_date'] ?? date('Y-m-01') ?>";&g...
42                 </div>
43                 <div class="form-group">
44                     <label>Fecha Fin</label>
45                     <input type="date" name="end_date" value="<?=$_GET['end_date'] ?? date('Y-m-d') ?>";&g...
46                 </div>
47                 <button type="submit">Filtrar</button>
48             </form>
49         </div>
50
51         <div class="stats-grid">
52             <div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
53                 <div class="stat-number"><?=$_stats['overall']['total_records'] ?></div>
54                 <div class="stat-label">Registros Totales</div>
55             </div>
56             <div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
57                 <div class="stat-number"><?=$_stats['overall']['presente'] ?></div>
58                 <div class="stat-label">Presentes</div>
59             </div>
60             <div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
61                 <div class="stat-number"><?=$_stats['overall']['ausente'] ?></div>
62                 <div class="stat-label">Ausentes</div>
63             </div>
64             <div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
65                 <div class="stat-number"><?=$_stats['overall']['tardanza'] ?></div>
66                 <div class="stat-label">Tardanzas</div>
67             </div>
68         </div>
69
70         <div class="card">
71             <h2>Asistencia por Curso</h2>
72             <table>
73                 <thead>
74                     <tr>
75                         <th>Curso</th>
```

```

76         <th>Total</th>;
77         <th>Presentes</th>;
78         <th>Ausentes</th>;
79         <th>% Asistencia</th>;
80         <th>Indicador</th>;
81     </tr>;
82 </thead>;
83 <tbody>;
84     <?php foreach($stats['by_course'] as $course): ?>;
85         <tr>;
86             <td><?=$course['course_name'] ?></td>;
87             <td><?=$course['total'] ?></td>;
88             <td><?=$course['presente'] ?></td>;
89             <td><?=$course['ausente'] ?></td>;
90             <td><?=$course['percentage'] ?>%</td>;
91             <td>;
92             <div class="progress-bar">;
93                 <div class="progress-fill" style="width: <?=$course['percentage'] ?>%;"><?=<?></div>;
94             </div>;
95         </td>;
96     </tr>;
97 <?php endforeach; ?>;
98 </tbody>;
99 </table>;
100 </div>;
101
102 <div class="card">;
103     <h2>Top 10 - Estudiantes con Más Ausencias</h2>;
104     <table>;
105         <thead>;
106             <tr>;
107                 <th>#</th>;
108                 <th>Estudiante</th>;
109                 <th>Curso</th>;
110                 <th>Total Ausencias</th>;
111             </tr>;
112         </thead>;
113         <tbody>;
114             <?php foreach($stats['by_student'] as $index =>$student): ?>;
115                 <tr>;
116                     <td><?=$index + 1 ?></td>;
117                     <td><?=$student['student_name'] ?></td>;
118                     <td><?=$student['course_name'] ?></td>;
119                     <td><strong style="color: #dc3545;"><?=$student['total_absences'] ?></stro...
120                 </tr>;
121             <?php endforeach; ?>;
122         </tbody>;
123     </table>;
124 </div>;
125
126 <div class="card">;
127     <h2>Tendencia Diaria</h2>;
128     <table>;
129         <thead>;
130             <tr>;
131                 <th>Fecha</th>;
132                 <th>Registros</th>;
133                 <th>Presentes</th>;
134                 <th>% Asistencia</th>;
135             </tr>;
136         </thead>;
137         <tbody>;
138             <?php foreach($stats['by_day'] as $day): ?>;
139                 <tr>;
140                     <td><?=$date('d/m/Y', strtotime($day['date'])) ?></td>;
141                     <td><?=$day['total'] ?></td>;
142                     <td><?=$day['presente'] ?></td>;
143                     <td><?=$day['percentage'] ?>%</td>;
144                 </tr>;
145             <?php endforeach; ?>;
146         </tbody>;
147     </table>;
148 </div>;
149 </div>;
150 </body>;
151 </html>;

```

## 81. course\_attendance.php

Ruta: 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:...
27         .breadcrumb a { color: #007bff; text-decoration: none; }
28         .container { max-width: 1200px; margin: 24px auto; padding: 0 16px; }
29         .course-header { background: linear-gradient(135deg, #007bff, #0056b3); color: #fff; border-radius: 10px; padding: 20...
30         .course-header .icon { font-size: 2.2rem; }
31         .course-header h1 { font-size: 1.3rem; font-weight: 700; }
32         .course-header p { font-size: 0.85rem; opacity: 0.85; margin-top: 4px; }
33
34         /* Stats */
35         #stats-section { display: flex; gap: 12px; margin-bottom: 20px; flex-wrap: wrap; }
36         .stat-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 14px 18px; flex: 1; min-width: ...
37         .stat-card .number { font-size: 1.8rem; font-weight: 700; }
38         .stat-card .label { font-size: 0.75rem; color: #888; margin-top: 2px; text-transform: uppercase; letter-spacing: 0.5...
39         .stat-card .total { border-top-color: #6c757d; } .stat-card .total .number { color: #6c757d; }
40         .stat-card .presente { border-top-color: #28a745; } .stat-card .presente .number { color: #28a745; }
41         .stat-card .ausente { border-top-color: #dc3545; } .stat-card .ausente .number { color: #dc3545; }
42         .stat-card .tardanza { border-top-color: #ffc107; } .stat-card .tardanza .number { color: #e6ac00; }
43         .stat-card .justif { border-top-color: #17a2b8; } .stat-card .justif .number { color: #17a2b8; }
44         .progress-bar { height: 8px; background: #e9ecef; border-radius: 4px; overflow: hidden; margin-top: 6px; }
45         .progress-fill { height: 100%; border-radius: 4px; transition: width 0.4s ease; }
46
47         /* Top ausencias */
48         .top-absences { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 16px; margin-bottom: 20px; ...
49         .top-absences h3 { font-size: 0.9rem; font-weight: 600; margin-bottom: 12px; color: #dc3545; }
50         .absence-item { display: flex; align-items: center; gap: 10px; padding: 6px 0; border-bottom: 1px solid #f0f0f0; }
51         .absence-item:last-child { border-bottom: none; }
52         .absence-name { flex: 1; font-size: 0.85rem; }
53         .absence-count { font-weight: 700; color: #dc3545; font-size: 0.95rem; min-width: 28px; text-align: right; }
54         .absence-bar { flex: 2; }
55
56         /* Filtros */
57         .filters { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 16px; margin-bottom: 16px; }
58         .filters h3 { font-size: 0.85rem; font-weight: 600; color: #555; margin-bottom: 12px; }
59         .filter-grid { display: grid; grid-template-columns: repeat(auto-fill, minmax(180px, 1fr)); gap: 10px; }
60         .filter-grid select,
61         .filter-grid input { width: 100%; padding: 7px 10px; border: 1px solid #ccc; border-radius: 6px; font-size: 0.85rem; ...
62         .filter-grid select:focus,
63         .filter-grid input:focus { border-color: #007bff; outline: none; box-shadow: 0 0 2px rgba(0, 123, 255, 0.15); }
64         .filter-grid select.on,
65         .filter-grid input.on { border-color: #007bff; background: #f0f7ff; }
66         .filter-banner { background: #fff3cd; border: 1px solid #ffc107; border-radius: 6px; padding: 7px 12px; font-size: 0...
67         .filter-banner a { color: #856404; font-weight: 700; text-decoration: none; cursor: pointer; }
68         .btn { padding: 7px 16px; border-radius: 6px; border: none; cursor: pointer; font-size: 0.85rem; font-weight: 500; tex...
69         .btn:hover { opacity: 0.85; }
70         .btn-outline { background: #fff; color: #555; border: 1px solid #ccc; }
71
72         /* Tabla */
73         .table-section { position: relative; }
74         .loading-overlay { display: none; position: absolute; inset: 0; background: rgba(255, 255, 255, 0.75); border-radius: 8p...
75         .loading-overlay.on { display: flex; }
```

```

76     .spinner { width:28px; height:28px; border:3px solid #e0e0e0; border-top-color:#007bff; border-radius:50%; anima...
77     @keyframes spin { to { transform:rotate(360deg); } }
78     .table-wrap { background:#fff; border:1px solid #e0e0e0; border-radius:8px; overflow:hidden; }
79     .table-info { padding:12px 16px; border-bottom:1px solid #e0e0e0; font-size:0.83rem; color:#666; display:flex; j...
80     table { width:100%; border-collapse:collapse; }

81     thead th { background:#f8f9fa; padding:10px 12px; text-align:left; font-size:0.8rem; font-weight:600; color:#555...
82     tbody td { padding:9px 12px; border-bottom:1px solid #f0f0f0; font-size:0.85rem; }
83     tbody tr:hover { background:#f8f9fa; }
84     tbody tr:last-child td { border-bottom:none; }
85     .status-badge { display:inline-block; padding:2px 10px; border-radius:12px; font-size:0.75rem; font-weight:600; ...
86     .empty-state { text-align:center; padding:50px 20px; color:#999; }
87     .empty-state .icon { font-size:2.5rem; margin-bottom:10px; }
88     </style>
89 </head>
90 <body>
91
92 <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
93
94 <div class="breadcrumb">
95     <a href="?action=dashboard">Inicio</a> &rsaquo;
96     <a href="?action=dashboard">Mi Dashboard</a> &rsaquo;
97     Asistencia de Mi Curso
98 </div>
99
100 <div class="container">
101
102     <div class="course-header">
103         <div class="icon"></div>
104         <div>
105             <h1>Asistencia - <?php echo htmlspecialchars($course['name']); </h1>
106             <p>
107                 Jornada: <?php echo htmlspecialchars($course['shift_name']); &nbsp;&nbsp;&nbsp;
108                 Nivel: <?php echo htmlspecialchars($course['grade_level']); &nbsp;&nbsp;&nbsp;
109                 Paralelo: <?php echo htmlspecialchars($course['parallel']); &nbsp;&nbsp;&nbsp;
110                 Todas las asignaturas
111             </p>
112         </div>
113     </div>
114
115     <!-- Stats - actualizado por AJAX -->
116     <div id="stats-section">
117         <?php if ($stats & $stats['total'] > 0): ?>
118             <div class="stat-card total"><div class="number"><?php echo $stats['total']; </div><div cla...
119             <div class="stat-card presente">
120                 <div class="number"><?php echo $stats['presente']; </div>
121                 <div class="label">Presentes (<?php echo $pct($stats['presente'], $stats['total']); </div>
122                 <div class="progress-bar"><div class="progress-fill" style="width:<?php echo $pct($stats['presente'], $...
123             </div>
124             <div class="stat-card ausente">
125                 <div class="number"><?php echo $stats['ausente']; </div>
126                 <div class="label">Ausentes (<?php echo $pct($stats['ausente'], $stats['total']); </div>
127                 <div class="progress-bar"><div class="progress-fill" style="width:<?php echo $pct($stats['ausente'], $s...
128             </div>
129             <div class="stat-card tardanza">
130                 <div class="number"><?php echo $stats['tardanza']; </div>
131                 <div class="label">Tardanzas (<?php echo $pct($stats['tardanza'], $stats['total']); </div>
132                 <div class="progress-bar"><div class="progress-fill" style="width:<?php echo $pct($stats['tardanza'], $...
133             </div>
134             <div class="stat-card justif">
135                 <div class="number"><?php echo $stats['justificado']; </div>
136                 <div class="label">Justificados (<?php echo $pct($stats['justificado'], $stats['total']); </div>
137                 <div class="progress-bar"><div class="progress-fill" style="width:<?php echo $pct($stats['justificado'], $...
138             </div>
139             <?php endif; ?>
140         </div>
141
142     <!-- Top ausencias (estático) -->
143     <?php if (!empty($topAbsences)): ?>
144     <div class="top-absences">
145         <h3>Top estudiantes con más ausencias</h3>
146         <?php $maxAbs = $topAbsences[0]['total_ausencias']; ?>
147         <foreach ($topAbsences as $i => $row): ?>
148             <div class="absence-item">
149                 <span style="font-size:0.75rem;color:#999;min-width:18px;"><?php echo $i+1 </span>
150                 <span class="absence-name"><?php echo htmlspecialchars($row['student_name']); </span>
151                 <div class="absence-bar"><div class="progress-bar"><div class="progress-fill" style="width:<?php echo $pct($row['total_ausencias'], $maxAbs); </div>
152                 <span class="absence-count"><?php echo $row['total_ausencias']; </span>
153             </div>
154         <?php endforeach; ?>

```

```

155 </div>
156 <?php endif; ?>
157
158 <!-- Filtros - con name para fallback, interceptados por JS -->
159 <div class="filters">
160     <h3>■ Filtrar asistencias</h3>
161
162     <div id="filter-banner" style="display:<? $hasFilters ? 'flex' : 'none' ?>;" class="filter-banner">
163         <span>■ Filtros activos</span>
164         <a onclick="clearFilters()">✕ Limpiar filtros</a>
165     </div>
166     <div class="filter-grid">
167         <div>
168             <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Asignatura</label>
169             <select name="subject_id" id="f_subject" onchange="runFilter()" class="<? !empty($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             </div>
177             <div>
178                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estudiante</label>
179                 <select name="student_id" id="f_student" onchange="runFilter()" class="<? !empty($filters['studen...
180                     <option value="">- Todos -</option>
181                     <?php foreach ($students as $st): ?>
182                         <option value="<? $st['id'] ?>" <? ($filters['student_id'] ?? '') == $st['id'] ? '...
183                             <? = htmlspecialchars($st['full_name']) ?>
184                         </option>
185                     <?php endforeach; ?>
186                 </div>
187             <div>
188                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estado</label>
189                 <select name="status" id="f_status" onchange="runFilter()" class="<? !empty($filters['status']) ?...
190                     <option value="">- Todos -</option>
191                     <option value="presente" <? ($filters['status'] ?? '') == 'presente' ? 'selected' : '...
192                     <option value="ausente" <? ($filters['status'] ?? '') == 'ausente' ? 'selected' : '...
193                     <option value="tardanza" <? ($filters['status'] ?? '') == 'tardanza' ? 'selected' : '...
194                     <option value="justificado" <? ($filters['status'] ?? '') == 'justificado' ? 'selected' : '...
195                 </select>
196             </div>
197             <div>
198                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Desde</label>
199                 <input type="date" name="start_date" id="f_start" onchange="runFilter()"
200                     value="<? htmlspecialchars($filters['start_date'] ?? '') ?>"
201                     class="<? !empty($filters['start_date']) ? 'on' : '' ?>">
202             </div>
203             <div>
204                 <label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Hasta</label>
205                 <input type="date" name="end_date" id="f_end" onchange="runFilter()"
206                     value="<? htmlspecialchars($filters['end_date'] ?? '') ?>"
207                     class="<? !empty($filters['end_date']) ? 'on' : '' ?>">
208             </div>
209         </div>
210     </div>
211 </div>
212
213 <!-- Tabla - actualizada por AJAX -->
214 <div class="table-section">
215     <div class="loading-overlay" id="loader"><div class="spinner"></div></div>
216     <div class="table-wrap">
217         <div class="table-info">
218             <span>■ <strong id="result-count"><? count($attendances) ?></strong> registros ...
219             <span id="filter-tag" style="color:#856404;font-size:0.8rem;display:<? $hasFilters ? 'inline' : '...
220         </div>
221         <div id="table-body">
222             <?php if (empty($attendances)): ?>
223                 <div class="empty-state"><div class="icon">■</div><p>No hay registros de asi...
224             <?php else: ?>
225                 <table>
226                     <thead>
227                         <tr>
228                             <th>Fecha</th>
229                             <th>Hora</th>
230                             <th>Estudiante</th>
231                             <th>Asignatura</th>
232                             <th>Docente</th>
233                             <th>Estado</th>
234                             <th>...
235                         </tr>
236                     </thead>
237                     <tbody>
238                         <?php foreach ($attendances as $a):
239                             $sl = $statusLabels[$a['status']] ?? ['label'=>$a['status'],'color'=>'#666','bg'=>'#eee...
240                             <tr>
241                                 <td><? $a['date'] ?></td>
242                                 <td><? $a['time'] ?></td>
243                                 <td><? $a['student'] ?></td>
244                                 <td><? $a['subject'] ?></td>
245                                 <td><? $a['teacher'] ?></td>
246                                 <td><? $a['status'] ?></td>
247                                 <td><? $a['details'] ?></td>
248                             </tr>
249                         </foreach>
250                     </tbody>
251                 </table>
252             </div>
253         </div>
254     </div>
255 </div>

```

[illegible]

```

313 function renderStats(s, total) {
314     var sec = document.getElementById('stats-section');
315     if (!s) return;
316     var t = parseInt(total) || 0;
317     var pre = parseInt(s.presente) || 0;
318     var aus = parseInt(s.ausente) || 0;
319     var tar = parseInt(s.tardanza) || 0;
320     var jus = parseInt(s.justificado) || 0;

321     function pct(v){ return t &gt; 0 ? Math.round(v/t*100)/10 : 0; }
322     function bar(v,c){ return '<div class="progress-bar"&gt;&lt;div class="progress-fill" style="width: '+pct(v)+'%;ba...
323     sec.innerHTML =
324         '<div class="stat-card total"&gt;&lt;div class="number"&gt;'+t+'&lt;/div&gt;&lt;div class="label"&gt;Total re...
325         '<div class="stat-card presente"&gt;&lt;div class="number"&gt;'+pre+'&lt;/div&gt;&lt;div class="label"&gt;Pre...
326         '<div class="stat-card ausente"&gt;&lt;div class="number"&gt;'+aus+'&lt;/div&gt;&lt;div class="label"&gt;Ause...
327         '<div class="stat-card tardanza"&gt;&lt;div class="number"&gt;'+tar+'&lt;/div&gt;&lt;div class="label"&gt;Tar...
328         '<div class="stat-card justif"&gt;&lt;div class="number"&gt;'+jus+'&lt;/div&gt;&lt;div class="label"&gt;Justi...
329     }
330
331 function renderTable(rows) {
332     document.getElementById('result-count').textContent = rows.length;
333     if (!rows.length) {
334         document.getElementById('table-body').innerHTML =
335             '<div class="empty-state"&gt;&lt;div class="icon"&gt;■&lt;/div&gt;&lt;p&gt;No se encontraron registros co...
336         return;
337     }
338     var html = '<table&gt;&lt;thead&gt;&lt;tr&gt;&lt;th&gt;Fecha&lt;/th&gt;&lt;th&gt;Hora&lt;/th&gt;&lt;th&gt;Estudia...
339     rows.forEach(function(a){
340         var sl = SL[a.status] || { label:a.status, color:'#666', bg:'#eee' };
341         var d = a.date ? a.date.substr(8,2)+'/'+a.date.substr(5,2)+'/'+a.date.substr(0,4) : '';
342         html += '<tr&gt;' +
343             '<td&gt;'+d+'&lt;/td&gt;' +
344             '<td style="white-space:nowrap;"&gt;'+esc(a.hour_period)+'&lt;/td&gt;' +
345             '<td&gt;'+esc(a.student_name)+'&lt;/td&gt;' +
346             '<td&gt;'+esc(a.subject_name)+'&lt;/td&gt;' +
347             '<td style="color:#666;"&gt;'+esc(a.teacher_name)+'&lt;/td&gt;' +
348             '<td&gt;&lt;span class="status-badge" style="background: '+sl.bg+';color: '+sl.color+';"&gt;'+sl.label+'&lt;...
349             '<td style="color:#888;font-size:0.82rem;"&gt;'+(a.observation ? esc(a.observation) : '-')+'&lt;/td&gt;' ...
350             '</tr&gt;';
351     });
352     html += '</tbody&gt;&lt;/table&gt;';
353     document.getElementById('table-body').innerHTML = html;
354 }
355
356 function esc(s){
357     if (!s) return '';
358     return String(s).replace(/&/g, '&amp;').replace(/&lt;/g, '&amp;lt;').replace(/&gt;/g, '&amp;gt;').replace(/"/g, ...
359 }
360
361 function clearFilters(){
362     ['f_subject', 'f_student', 'f_status'].forEach(function(id){ document.getElementById(id).value = ''; });
363     document.getElementById('f_start').value = '';
364     document.getElementById('f_end').value = '';
365     runFilter();
366 }
367 &lt;/script&gt;
368
369 &lt;/body&gt;
370 &lt;/html&gt;

```

## 82. no\_tutor.php

Ruta: ArchImp/views/tutor/no\_tutor.php

```
1  <?php
2  // views/tutor/no_tutor.php
3  ?>
4  <!DOCTYPE html>
5  <html lang="es">
6  <head><meta charset="UTF-8"><title>EcuAsist</title></head>
7  <body>
8  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
9  <div style="max-width:500px;margin:80px auto;text-align:center;font-family:sans-serif;color:#666;">
10     <div style="font-size:3rem;">■■■■</div>
11     <h2 style="margin:16px 0 8px;">No eres tutor de ningún curso</h2>
12     <p>Actualmente no tienes asignado un curso como tutor en el año lectivo activo.</p>
13     <a href="?action=dashboard" style="display:inline-block;margin-top:20px;padding:10px 20px;background:#007bff;colo...
14 </div>
15 </body>
16 </html>
```



## 83. create.php

**Ruta:** ArchImp/views/users/create.php

[illegible]

```

76         value="&lt;? isset($_POST['email']) ? htmlspecialchars($_POST['email']) ...
77         required&gt;
78     &lt;/div&gt;
79     &lt;div class="col-md-6 mb-3"&gt;
80         &lt;label class="form-label required"&gt;Contraseña&lt;/label&gt;
81
82         &lt;input type="password"
83             name="password"
84             class="form-control"
85             minlength="6"
86             required&gt;
87         &lt;small class="text-muted"&gt;Mínimo 6 caracteres&lt;/small&gt;
88     &lt;/div&gt;
89     &lt;div class="col-md-6 mb-3"&gt;
90         &lt;label class="form-label required"&gt;Confirmar Contraseña&lt;/label&gt;
91         &lt;input type="password"
92             name="confirm_password"
93             class="form-control"
94             minlength="6"
95             required&gt;
96     &lt;/div&gt;
97 &lt;/div&gt;
98
99 &lt;!-- Sección Datos Personales --&gt;
100 &lt;div class="form-section"&gt;
101     &lt;h5&gt;■■■ Datos Personales&lt;/h5&gt;
102     &lt;div class="row"&gt;
103         &lt;div class="col-md-6 mb-3"&gt;
104             &lt;label class="form-label required"&gt;Nombres&lt;/label&gt;
105             &lt;input type="text"
106                 name="first_name"
107                 class="form-control"
108                 value="&lt;? isset($_POST['first_name']) ? htmlspecialchars($_POST['firs...
109                 required&gt;
110         &lt;/div&gt;
111         &lt;div class="col-md-6 mb-3"&gt;
112             &lt;label class="form-label required"&gt;Apellidos&lt;/label&gt;
113             &lt;input type="text"
114                 name="last_name"
115                 class="form-control"
116                 value="&lt;? isset($_POST['last_name']) ? htmlspecialchars($_POST['last...
117                 required&gt;
118         &lt;/div&gt;
119         &lt;div class="col-md-6 mb-3"&gt;
120             &lt;label class="form-label"&gt;Cédula/DNI&lt;/label&gt;
121             &lt;input type="text"
122                 name="dni"
123                 class="form-control"
124                 value="&lt;? isset($_POST['dni']) ? htmlspecialchars($_POST['dni']) : '...'
125         &lt;/div&gt;
126         &lt;div class="col-md-6 mb-3"&gt;
127             &lt;label class="form-label"&gt;Teléfono&lt;/label&gt;
128             &lt;input type="text"
129                 name="phone"
130                 class="form-control"
131                 value="&lt;? isset($_POST['phone']) ? htmlspecialchars($_POST['phone']) ...
132         &lt;/div&gt;
133     &lt;/div&gt;
134 &lt;/div&gt;
135
136 &lt;!-- Sección Roles --&gt;
137 &lt;div class="form-section"&gt;
138     &lt;h5&gt;■ Roles del Usuario&lt;/h5&gt;
139     &lt;p class="text-muted"&gt;Selecciona uno o varios roles (opcional, puedes asignarlos d...
140     &lt;div class="row"&gt;
141         &lt;?php foreach ($roles as $role): ?&gt;
142             &lt;div class="col-md-4 mb-2"&gt;
143                 &lt;div class="form-check"&gt;
144                     &lt;input class="form-check-input"
145                         type="checkbox"
146                         name="roles[]"
147                         value="&lt;? $role['id'] ?&gt;"
148                         id="role_&lt;? $role['id'] ?&gt;"
149                     &lt;? isset($_POST['roles']) &amp; in_array($role['id'], $_...
150                     &lt;label class="form-check-label" for="role_&lt;? $role['id'] ?&gt;"&g...
151                         &lt;? ucfirst($role['name']) ?&gt;
152                     &lt;/label&gt;
153                 &lt;/div&gt;
154             &lt;/div&gt;

```

```

155             &lt;?php endforeach; ?&gt;
156         &lt;/div&gt;
157     &lt;/div&gt;
158
159     &lt;!-- Botones --&gt;
160     &lt;div class="d-flex justify-content-between mt-4"&gt;
161
162         &lt;a href="?action=users" class="btn btn-secondary"&gt;
163             ← Cancelar
164         &lt;/a&gt;
165         &lt;button type="submit" class="btn btn-primary"&gt;
166             ✓ Crear Usuario
167         &lt;/button&gt;
168     &lt;/div&gt;
169     &lt;/form&gt;
170 &lt;/div&gt;
171 &lt;/div&gt;
172 &lt;/div&gt;
173 &lt;/div&gt;
174
175     &lt;script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
176 &lt;/body&gt;
177 &lt;/html&gt;

```

## 84. edit.php

Ruta: ArchImp/views/users/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 Usuario - 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         .password-info {
34             background: #fff3cd;
35             border-left: 4px solid #ffc107;
36             padding: 10px;
37             margin-bottom: 15px;
38         }
39     </style>
40 </head>
41 <body>
42     <?php include BASE_PATH . '/views/partials/navbar.php'; ?>
43
44     <div class="container">
45         <div class="row justify-content-center">
46             <div class="col-md-10">
47                 <div class="card shadow">
48                     <div class="card-header bg-primary text-white">
49                         <h4 class="mb-0">■ Editar Usuario: <?php echo htmlspecialchars($user['first_name']) . ' ' . $u...
50                     </div>
51                     <div class="card-body">
52                         <?php if (!empty($errors)): ?>
53                             <div class="alert alert-danger">
54                                 <strong>Errores:</strong>
55                                 <ul class="mb-0">
56                                     <?php foreach ($errors as $error): ?>
57                                         <li><?php echo htmlspecialchars($error) ?></li>
58                                     <?php endforeach; ?>
59                                 </ul>
60                             </div>
61                         <?php endif; ?>
62
63                         <form method="POST" action="?action=edit_user&id=? $user['id'] ?>">
64                             <!-- Sección Datos de Acceso -->
65                             <div class="form-section">
66                                 <h5>■ Datos de Acceso</h5>
67                                 <div class="row">
68                                     <div class="col-md-6 mb-3">
69                                         <label class="form-label required">Nombre de Usuario</label>
70                                         <input type="text"
71                                             name="username"
72                                             class="form-control"
73                                             value="<?php echo htmlspecialchars($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
82                 class="form-control"
83                 value="{htmlspecialchars($user['email'])}"
84                 required>
85         </div>
86     </div>
87     <div class="password-info">
88         <strong>■■■ Contraseña: </strong> Deja en blanco si no deseas cambiarla
89     </div>
90
91     <div class="row">
92         <div class="col-md-6 mb-3">
93             <label class="form-label">Nueva Contraseña</label>
94             <input type="password"
95                 name="password"
96                 class="form-control"
97                 minlength="6">
98             <small class="text-muted">Mínimo 6 caracteres</small>
99         </div>
100        <div class="col-md-6 mb-3">
101            <label class="form-label">Confirmar Nueva Contraseña</label>
102            <input type="password"
103                name="confirm_password"
104                class="form-control"
105                minlength="6">
106        </div>
107    </div>
108 </div>
109
110 <!-- Sección Datos Personales -->
111 <div class="form-section">
112     <h5>■■■ Datos Personales</h5>
113     <div class="row">
114         <div class="col-md-6 mb-3">
115             <label class="form-label required">Nombres</label>
116             <input type="text"
117                 name="first_name"
118                 class="form-control"
119                 value="{htmlspecialchars($user['first_name'])}"
120                 required>
121         </div>
122         <div class="col-md-6 mb-3">
123             <label class="form-label required">Apellidos</label>
124             <input type="text"
125                 name="last_name"
126                 class="form-control"
127                 value="{htmlspecialchars($user['last_name'])}"
128                 required>
129         </div>
130         <div class="col-md-6 mb-3">
131             <label class="form-label">Cédula/DNI</label>
132             <input type="text"
133                 name="dni"
134                 class="form-control"
135                 value="{htmlspecialchars($user['dni'])}">
136         </div>
137         <div class="col-md-6 mb-3">
138             <label class="form-label">Teléfono</label>
139             <input type="text"
140                 name="phone"
141                 class="form-control"
142                 value="{htmlspecialchars($user['phone'])}">
143         </div>
144     </div>
145 </div>
146
147 <!-- Información de Roles -->
148 <div class="alert alert-info">
149     <strong>■■■ Roles: </strong> Los roles se gestionan desde la tabla de usuarios...
150 </div>
151
152 <!-- Botones -->
153 <div class="d-flex justify-content-between mt-4">
154     <a href="{action=users}" class="btn btn-secondary">

```

```
155             ← Cancelar
156             &lt;/a&gt;
157             &lt;button type="submit" class="btn btn-success"&gt;
158                 ✓ Guardar Cambios
159             &lt;/button&gt;
160         &lt;/div&gt;

161         &lt;/form&gt;
162     &lt;/div&gt;
163 &lt;/div&gt;
164 &lt;/div&gt;
165 &lt;/div&gt;
166 &lt;/div&gt;
167
168     &lt;script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
169 &lt;/body&gt;
170 &lt;/html&gt;
```

## 85. index.php

Ruta: 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: #c82323; }
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
71     <div class="container">
72         <div class="card">
73             <?php if(isset($_GET['created'])): ?>
74                 <div class="success">✓ Usuario creado correctamente</div>
75             <?php endif; ?>
```

```

76     <?php if(isset($_GET['updated'])): ?>
77         <div class="success"><img alt="checkmark" /> Usuario actualizado correctamente</div>
78     <?php endif; ?>
79     <?php if(isset($_GET['deleted'])): ?>
80         <div class="success"><img alt="checkmark" /> Usuario eliminado correctamente</div>
81     <?php endif; ?>
82     <?php if(isset($_GET['deactivated'])): ?>
83         <div class="info"><img alt="info icon" /> Usuario desactivado (tiene registros de asistencia)</div>
84     <?php endif; ?>
85     <?php if(isset($_GET['success'])): ?>
86         <div class="success"><img alt="checkmark" /> Rol asignado correctamente</div>
87     <?php endif; ?>
88     <?php if(isset($_GET['removed'])): ?>
89         <div class="success"><img alt="checkmark" /> Rol eliminado correctamente</div>
90     <?php endif; ?>
91     <?php if(isset($_GET['error']) && $_GET['error'] === 'has_assignments'): ?>
92         <div class="error"><img alt="error icon" /> No se puede eliminar el rol docente porque tiene asignaciones activas. Elimin...
93     <?php endif; ?>
94     <?php if(isset($_GET['error']) && $_GET['error'] === 'not_found'): ?>
95         <div class="error"><img alt="error icon" /> Usuario no encontrado</div>
96     <?php endif; ?>
97     <?php if(isset($_GET['error']) && $_GET['error'] === 'self_delete'): ?>
98         <div class="error"><img alt="error icon" /> No puedes eliminar tu propia cuenta</div>
99     <?php endif; ?>
100    <?php if(isset($_GET['error']) && $_GET['error'] === 'delete_failed'): ?>
101        <div class="error"><img alt="error icon" /> Error al eliminar el usuario</div>
102    <?php endif; ?>
103
104    <div class="header-actions">
105        <h2>Gestión de Usuarios</h2>
106        <a href="?action=create_user" style="padding: 10px 20px; background: #28a745; color: white; text-deco...
107            + Crear Usuario
108        </a>
109    </div>
110
111    <!-- Filtro por Rol -->
112    <form method="GET" action="" style="margin-bottom: 20px;">
113        <input type="hidden" name="action" value="users">
114        <div style="display: flex; gap: 15px; align-items: flex-end;">
115            <div style="flex: 1;">
116                <label style="display: block; margin-bottom: 5px; font-weight: bold; color: #333;">Filtrar...
117                <select name="filter_role" onchange="this.form.submit()" style="width: 100%; padding: 10px; b...
118                    <option value="">Todos los roles</option>
119                    <?php foreach($roles as $role): ?>
120                        <option value="<?php echo $role['name']; ?>" <?php echo (isset($_GET['filter_role']) && ...
121                            <?php echo ucfirst($role['name']); ?>
122                    </option>
123                <?php endforeach; ?>
124            </select>
125        </div>
126        <?php if(isset($_GET['filter_role']) && $_GET['filter_role'] != ''): ?>
127            <div>
128                <a href="?action=users" style="padding: 10px 15px; background: #6c757d; color: white; text-de...
129                    Limpiar
130                </a>
131            </div>
132        <?php endif; ?>
133    </div>
134    </form>
135
136    <table>
137        <thead>
138            <tr>
139                <th><img alt="refresh icon" /></th>
140                <th>Nombre Completo</th>
141                <th>Email</th>
142                <th>Roles Actuales</th>
143                <th>Asignar Rol</th>
144                <th>Acciones</th>
145            </tr>
146        </thead>
147        <tbody>
148            <?php
149                // Aplicar filtro
150                $filteredUsers = $users;
151
152                if(isset($_GET['filter_role']) && $_GET['filter_role'] != '') {
153                    $filterRole = $_GET['filter_role'];
154                    $filteredUsers = array_filter($users, function($user) use ($filterRole) {

```



```

155         if($user['roles']) {
156             $userRoles = explode(',', $user['roles']);
157             return in_array($filterRole, $userRoles);
158         }
159         return false;
160     });
161 }
162 ?>
163
164 <?php if(empty($filteredUsers)): ?>
165     <tr>
166         <td colspan="7" style="text-align:center;">No hay usuarios con este rol</td>
167     </tr>
168 <?php else: ?>
169     <?php
170         $counter = 1;
171         foreach($filteredUsers as $user):
172             ?>
173             <tr>
174                 <td><?php if($user['roles']): ?>
175                     <td><?php
176                         // Obtener IDs de roles con nombres
177                         $db = new Database();
178                         $sqlRoles = "SELECT r.id, r.name
179                                     FROM roles r
180                                     INNER JOIN user_roles ur ON r.id = ur.role_id
181                                     WHERE ur.user_id = :user_id
182                                     ORDER BY r.name";
183                         $stmtRoles = $db->connect()->prepare($sqlRoles);
184                         $stmtRoles->execute([':user_id' => $user['id']]);
185                         $userRolesData = $stmtRoles->fetchAll(PDO::FETCH_ASSOC);
186
187                         foreach($userRolesData as $roleData):
188                             ?>
189                             <span class="badge">
190                                 <?php
191                                     $roleName = ucfirst($roleData['name']);
192                                     <form method="POST" action="?action=remove_role">
193                                         <input type="hidden" name="user_id" value="<?php $user['id'] ?>";
194                                         <input type="hidden" name="role_id" value="<?php $roleData['id'] ?>";
195                                         <button type="submit" class="btn-remove-role">X</button>
196                                     </form>
197                                 </span>
198                             <?php endforeach; ?>
199
200                             <form method="POST" action="?action=assign_role">
201                                 <input type="hidden" name="user_id" value="<?php $user['id'] ?>";
202                                 <select name="role_id" required style="padding: 5px;">
203                                     <option value="">Seleccionar...
204                                     <?php foreach($roles as $role): ?>
205                                         <option value="<?php $role['id'] ?>"><?php $role['name'] ?>
206                                     </select>
207                                     <button type="submit" style="padding: 5px 10px;">Asignar</button>
208                                 </form>
209                             </td>
210                         <td class="action-buttons">
211                             <a href="?action=edit_user&id=<?php $user['id'] ?>"
212                                 class="btn-warning btn-sm"
213                                 style="text-decoration: none; padding: 5px 10px; border-radius: 4px;">
214                                 ■ Editar
215                             </a>
216                             <form method="POST" action="?action=delete_user" style="display: inline;" onsubmit="r...
217                                 <input type="hidden" name="user_id" value="<?php $user['id'] ?>";
218                                 <button type="submit" class="btn-danger btn-sm">
219                                     ■ Eliminar
220                                 </button>
221                             </td>
222                         </tr>
223                     </td>
224                 </td>
225             </tr>
226             <?php endforeach; ?>
227         </td>
228     </tr>
229 <?php endif; ?>

```

```

234         </tbody>;
235     </table>;
236 </div>;
237 </div>;
238
239 <script>;
240 function confirmDelete(event, userName) {
241
242     event.preventDefault();
243
244     const modal = document.createElement('div');
245     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
246
247     const modalContent = document.createElement('div');
248     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
249
250     modalContent.innerHTML = `
251         <h3 style="margin: 0 0 15px 0; color: #dc3545;"><span style="color: red; font-weight: bold;"> Eliminar Usuario</h3>;
252         <p style="margin: 0 0 20px 0; color: #666;">;
253             ¿Está seguro de eliminar al usuario <strong>${userName}</strong>?</p>;
254         <p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">;
255             <strong>Nota:</strong> Si el usuario tiene asistencias registradas, será desactivado en lugar...
256         </p>;
257         <div style="display: flex; gap: 10px; justify-content: flex-end;">;
258             <button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: whi...
259                 Cancelar
260             </button>;
261             <button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: wh...
262                 Sí, Eliminar
263             </button>;
264         </div>;
265     `;
266
267     modal.appendChild(modalContent);
268     document.body.appendChild(modal);
269
270     const form = event.target;
271
272     document.getElementById('confirmDeleteBtn').onclick = function() {
273         document.body.removeChild(modal);
274         form.submit();
275     };
276
277     document.getElementById('cancelDeleteBtn').onclick = function() {
278         document.body.removeChild(modal);
279     };
280
281     return false;
282 }
283
284 function confirmRemoveRole(event, roleName, userName) {
285     event.preventDefault();
286
287     const modal = document.createElement('div');
288     modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5);...
289
290     const modalContent = document.createElement('div');
291     modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px; box-shadow...
292
293     modalContent.innerHTML = `
294         <h3 style="margin: 0 0 15px 0; color: #dc3545;"><span style="color: red; font-weight: bold;"> Eliminar Rol</h3>;
295         <p style="margin: 0 0 20px 0; color: #666;">;
296             ¿Está seguro de eliminar el rol <strong>${roleName}</strong> del usuario <strong>${use...
297         </p>;
298         <div style="display: flex; gap: 10px; justify-content: flex-end;">;
299             <button type="button" id="cancelRemoveRoleBtn" style="padding: 10px 20px; background: #6c757d; color: ...
300                 Cancelar
301             </button>;
302             <button type="button" id="confirmRemoveRoleBtn" style="padding: 10px 20px; background: #dc3545; color: ...
303                 Sí, Eliminar
304             </button>;
305         </div>;
306     `;
307
308     modal.appendChild(modalContent);
309     document.body.appendChild(modal);
310
311     const form = event.target;
312

```

```
313     document.getElementById('confirmRemoveRoleBtn').onclick = function() {
314         document.body.removeChild(modal);
315         form.submit();
316     };
317
318     document.getElementById('cancelRemoveRoleBtn').onclick = function() {
319         document.body.removeChild(modal);
320     };
321
322     return false;
323 }
324 </script>
325 </body>
326 </html>
```

## 86. FuncionalidadesRol.md

Ruta: 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 ahí 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 debería 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
322 - **Acción sugerida:**
323   - Si SÍ → Crear lógica en `JustificationController`
324   - Agregar permiso especial para tutores
325   - Limitar solo a justificaciones de SU curso
326
327 ##### 2. Docente Tutor - Rechazar justificaciones de su curso
328 - **Estado:** NO implementado
329 - **Pregunta:** ¿Debe implementarse?
330 - **Impacto:** Complementa la funcionalidad #1
331 - **Acción sugerida:**
332   - Si SÍ → Agregar al mismo controller
333   - Mismo permiso que aprobación
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:** ■ Dependiente
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
```

## 87. README.md

**Ruta:** README.md

```

1 yþ# ■E■c■u■A■s■i■s■t■e■n■c■i■a■2■0■2■6■
2 ■
3 ■#■ ■E■c■u■A■s■i■s■t■e■n■c■i■a■2■0■2■6■ ■-■ ■S■i■s■t■e■m■a■ ■d■e■ ■A■s■i■s■t■e■n■c■i■a■ ■E■s■c■o■l■a■r■
4 ■
5 ■
6 ■
7 ■S■i■s■t■e■m■a■ ■d■e■ ■g■e■s■t■i■ó■n■ ■d■e■ ■a■s■i■s■t■e■n■c■i■a■ ■e■s■c■o■l■a■r■ ■d■e■s■a■r■r■o■l■l■a■d■o■ ■e■n■
8 ■
9 ■
10 ■
11 ■#■#■ ■=0■þ ■C■a■r■a■c■t■e■r■í■s■t■i■c■a■s■
12 ■
13 ■
14 ■
15 ■#■#■#■ ■M■ó■d■u■l■o■s■ ■P■r■i■n■c■i■p■a■l■e■s■
16 ■
17 ■-■ ■' ■A■u■t■e■n■t■i■c■a■c■i■ó■n■ ■y■ ■r■o■l■e■s■ ■m■u■l■t■i■-■u■s■u■a■r■i■o■
18 ■
19 ■-■ ■' ■G■e■s■t■i■ó■n■ ■d■e■ ■u■s■u■a■r■i■o■s■ ■(■e■s■t■u■d■i■a■n■t■e■s■,■ ■d■o■c■e■n■t■e■s■,■ ■a■u■t■o■r■i■d■a■
20 ■
21 ■-■ ■' ■R■e■g■i■s■t■r■o■ ■d■e■ ■a■s■i■s■t■e■n■c■i■a■ ■c■o■n■ ■v■a■l■i■d■a■c■i■ó■n■ ■d■e■ ■f■e■c■h■a■s■
22 ■
23 ■-■ ■' ■S■i■s■t■e■m■a■ ■d■e■ ■j■u■s■t■i■f■i■c■a■c■i■o■n■e■s■ ■c■o■n■ ■a■p■r■o■b■a■c■i■ó■n■
24 ■
25 ■-■ ■' ■R■e■p■o■r■t■e■s■ ■e■n■ ■P■D■F■ ■y■ ■E■x■c■e■l■
26 ■
27 ■-■ ■' ■C■a■l■e■n■d■a■r■i■o■ ■d■e■ ■a■s■i■s■t■e■n■c■i■a■s■
28 ■
29 ■-■ ■' ■S■i■s■t■e■m■a■ ■d■e■ ■h■o■r■a■r■i■o■s■ ■p■o■r■ ■c■u■r■s■o■ ■(■d■í■a■/■h■o■r■a■/■m■a■t■e■r■i■a■/■d■o■c■e■
30 ■
31 ■-■ ■' ■C■o■n■f■i■g■u■r■a■c■i■ó■n■ ■d■e■ ■i■n■s■t■i■t■u■c■i■ó■n■ ■c■o■n■ ■p■r■o■v■i■n■c■i■a■s■/■c■i■u■d■a■d■e■s■
32 ■
33 ■-■ ■' ■G■e■s■t■i■ó■n■ ■d■e■ ■j■o■r■r■n■a■d■a■s■ ■p■o■r■ ■i■n■s■t■i■t■u■c■i■ó■n■ ■(■m■a■t■u■t■i■n■a■/■v■e■s■p■e■r■
34 ■
35 ■-■ ■' ■G■e■s■t■i■ó■n■ ■s■e■p■a■r■a■d■a■ ■d■e■ ■t■u■t■o■r■e■s■
36 ■
37 ■-■ ■' ■E■s■t■a■d■í■s■t■i■c■a■s■ ■a■v■a■n■z■a■d■a■s■
38 ■
39 ■-■ ■' ■A■s■i■g■n■a■c■i■o■n■e■s■ ■d■o■c■e■n■t■e■s■ ■y■ ■t■u■t■o■r■e■s■
40 ■
41 ■-■ ■' ■R■e■l■a■c■i■o■n■e■s■ ■r■e■p■r■e■s■e■n■t■a■n■t■e■-■e■s■t■u■d■i■a■n■t■e■
42 ■
43 ■-■ ■' ■S■i■s■t■e■m■a■ ■d■e■ ■r■e■s■p■a■l■d■o■s■
44 ■
45 ■-■ ■' ■L■o■g■s■ ■d■e■ ■a■c■t■i■v■i■d■a■d■
46 ■
47 ■
48 ■
49 ■#■#■#■ ■T■e■c■n■o■l■o■g■í■a■s■
50 ■
51 ■-■ ■P■H■P■ ■7■.■4■+■
52 ■
53 ■-■ ■M■y■S■Q■L■ ■5■.■7■+■ ■/■ ■M■a■r■i■a■D■B■
54 ■
55 ■-■ ■P■D■O■ ■p■a■r■a■ ■b■a■s■e■ ■d■e■ ■d■a■t■o■s■
56 ■
57 ■-■ ■P■H■P■M■a■i■l■e■r■ ■p■a■r■a■ ■c■o■r■r■e■o■s■
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 ■#■#■ ■=0■æÜ ■I■n■s■t■a■l■a■c■i■ó■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■ ■&gt;■=■ ■7■.■4■
74 ■
75 ■-■ ■M■y■S■Q■L■ ■&gt;■=■ ■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■.■g■i■  
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■g■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■c■i■6■  
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■\*■\*■ ■(■L■i■n■u■x■/■M■a■c■)■\*■\*■  
130 ■  
131 ■`■`■`■b■a■s■h■  
132 ■  
133 ■c■h■m■o■d■ ■-■R■ 77555 ■.■  
134 ■  
135 ■c■h■m■o■d■ ■-■R■ 77777 ■u■p■l■o■a■d■s■  
136 ■  
137 ■c■h■m■o■d■ ■-■R■ 77777 ■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 ■#■#■ ■=0dŭ ■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 ■Üs su a r r i o d e e p r u e b a ( d e s p u é s d e e i m p o r t a r S Q L d e p r u e b a ) : ■  
158 ■  
159 ■- ■ \* \* \* Ü s s u a r r i o : \* \* \* p r o f . g a r c i a ■  
160 ■  
  
161 ■- ■ \* \* \* C o n t r a s e ñ a : \* \* \* p a s s w o r d ■  
162 ■  
163 ■  
164 ■  
165 ■# ■# ■ = Ø Å Û E s t r u c t u r a d e l 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 ó 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 ■  
177 ■ % % % v i e w s / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # V i s t a s H T M L / P H P ■  
178 ■  
179 ■ % % % v i e w s / s c h e d u l e / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # G e s t i ó n d e h o r a r i o s ■  
180 ■  
181 ■ % % % v i e w s / i n s t i t u t i o n / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # C o n f i g i n s t i t u c i ó n ■  
182 ■  
183 ■ % % % u p l o a d s / i n s t i t u t i o n / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # L o g o s i n s t i t u c i o n a l e s ■  
184 ■  
185 ■ % % % h e l p e r s / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # U t i l i d a d e s ■  
186 ■  
187 ■ % % % p u b l i c / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # P u n t o d e e n t r a d a ■  
188 ■  
189 ■ % % % u p l o a d s / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # A r c h i v o s s u b i d o s ■  
190 ■  
191 ■ % % % b a c k u p s / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # R e s p a l d o s a u t o m á t i c o s ■  
192 ■  
193 ■ % % % v e n d o r / ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # D e p e n d e n c i a s C o m p o s e r ■  
194 ■  
195 ■ ` ` ` ■  
196 ■  
197 ■  
198 ■  
199 ■# ■# ■ = Ø Ý R o l e s y P e r m i s o s ■  
200 ■  
201 ■  
202 ■  
203 ■- ■ \* \* \* A u t o r i d a d : \* \* \* A c c e s o t o t a l l a l s i s t e m a ■  
204 ■  
205 ■- ■ \* \* \* D o c e n t e : \* \* \* R e g i s t r o d e a s i s t e n c i a , y v e r c u r s o s a s i g n a ■  
206 ■  
207 ■- ■ \* \* \* I n s p e c t o r : \* \* \* v e r a s i s t e n c i a s , a p r o b a r j u s t i f i c a c i o n e s ■  
208 ■  
209 ■- ■ \* \* \* E s t u d i a n t e : \* \* \* v e r s u p r o p i a a s i s t e n c i a , j u s t i f i c a r a ■  
210 ■  
211 ■- ■ \* \* \* R e p r e s e n t a n t e : \* \* \* v e r a s i s t e n c i a d e r e p r e s e n t a d o s ■  
212 ■  
213 ■  
214 ■  
215 ■# ■# ■ = Ø Ê Û R e p o r t e s ■  
216 ■  
217 ■  
218 ■  
219 ■- ■ \* \* \* P D F : \* \* \* F o r m a t o i n s t i t u c i o n a l c o n l o g o y e n c a b e z a d o s ■  
220 ■  
221 ■- ■ \* \* \* E x c e l : \* \* \* D a t o s e s t r u c t u r a d o s p a r a a n á l i s i s ■  
222 ■  
223 ■- ■ \* \* \* C a l l e n d a r i o : \* \* \* V i s t a m e n s u a l d e a s i s t e n c i a s ■  
224 ■  
225 ■- ■ \* \* \* E s t a d í s t i c a s : \* \* \* M é t r i c a s p o r c u r s o , e s t u d i a n t e y p e r ■  
226 ■  
227 ■  
228 ■  
229 ■# ■# ■ = Ø Ý V a l i d a c i o n e s i m p o r t a n t e s ■  
230 ■  
231 ■  
232 ■  
233 ■- ■ R e g i s t r o d e a s i s t e n c i a : m á x i m o 4 8 h o r a s h á b i l e s a t r á s ■

234 ■  
235 ■-■ N■o■ s■e■ p■e■r■m■i■t■e■ r■e■g■i■s■t■r■o■ d■e■ f■e■c■h■a■s■ f■u■t■u■r■a■s■  
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■á■l■c■u■l■o■ d■e■ 4■8■ h■o■r■a■s■  
238 ■  
239 ■-■ E■d■i■c■i■ó■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■é■s■  
240 ■  
  
241 ■  
242 ■  
243 ■#■#■ ■=0áp■p ■S■e■g■u■r■i■d■a■d■  
244 ■  
245 ■  
246 ■  
247 ■-■ C■o■n■t■r■a■s■e■ñ■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■ó■n■ C■S■R■F■ e■n■ f■o■r■m■u■l■a■r■i■o■s■  
250 ■  
251 ■-■ v■a■l■l■i■d■a■c■i■ó■n■ d■e■ e■n■t■r■a■d■a■s■  
252 ■  
253 ■-■ S■a■n■i■t■i■z■a■c■i■ó■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■e■s■  
256 ■  
257 ■  
258 ■  
259 ■#■#■ ■=0FÛ ■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 ■#■#■ ■=0ÃÛ ■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■ó■n■  
272 ■  
273 ■  
274 ■  
275 ■#■#■ ■=0'Ý ■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■á■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■á■t■i■c■a■ d■e■ &gt;300 d■i■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■ó■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■x■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■ó■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■á■f■i■c■o■s■ i■n■t■e■r■a■c■t■i■v■o■s■  
308 ■  
309 ■-■ {[■ ■]} ■B■ó■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 ■p■a■r■a ■i■n■s■t■i■t■u■c■i■o■n■e■s ■e■d■u■c■a■t■i■v■a■s ■e■c■u■a■t■o■

## 88. ResumenEjecutivo\_06C.md

Ruta: ResumenEjecutivo\_06C.md

```
1 # ■ RESUMEN EJECUTIVO – ECUASIST 2026 (Consolidado General Final)
2
3 **Fecha:** 18 de Febrero de 2026
4 **Versión:** v1.6
5
6 ---
7
8 ## ■ ESTADO DEL PROYECTO
9
10 Sistema integral de asistencia escolar desarrollado en **PHP OOP puro + MySQL**, bajo arquitectura MVC modular y sin fra...
11
12 - Módulos completados: **19/19**
13 - Bugs críticos: **0**
14 - Enfoque actual: Optimización, refinamiento visual y preparación productiva
15 - Estado: Listo para producción tras validación final de rutas, uploads y timezone
16
17 El sistema se encuentra en fase avanzada de optimización, no en construcción base.
18
19 ---
20
21 ## ■ MEJORAS IMPLEMENTADAS
22
23 ---
24
25 ### 1■■ Sistema de Horarios Académicos
26
27 - Tabla `class_schedule`
28 - Validación anti-duplicados curso/día/hora
29 - Detección de conflictos en tiempo real
30 - Auto-asignación docente
31 - Horas diferenciadas por nivel
32 - Gestión visual por curso
33
34 **Flujo operativo:**
35 1. Crear curso
36 2. Asignar docente-materia
37 3. Configurar horario
38 4. Visualización automática para docente
39
40 ---
41
42 ### 2■■ Registro de Asistencia Inteligente
43
44 - Eliminada selección manual de curso
45 - Detección automática de clases del día
46 - Interfaz tipo tarjetas
47 - Validación de 48h hábiles
48 - Actualización automática si existe registro
49 - Precarga de estados guardados
50
51 **Métodos clave:**
52 - `getScheduleInfo()`
53 - `getExistingAttendance()`
54
55 ---
56
57 ### 3■■ NUEVO MÓDULO: Docente Tutor – Asistencia de Mi Curso
58
59 Permite al tutor visualizar toda la asistencia de su curso con estadísticas dinámicas.
60
61 #### ■ Backend
62
63 **Attendance.php** – 6 métodos agregados:
64
65 - `getTutorCourseId($teacherId)`
66 - `getSubjectsByCourse($courseId)`
67 - `getStudentsByCourse($courseId)`
68 - `getTutorCourseAttendance($courseId, $filters)`
69 - `getTutorCourseStats($courseId, $filters)`
70 - `getTutorTopAbsences($courseId, $limit)`
71
72 **TutorController.php – NUEVO**
73 - `courseAttendance()` – Vista principal
74 - `ajax()` – Endpoint JSON para filtros dinámicos
75
```

```

76 ##### ■ Frontend
77
78 - Stats cards con barras de progreso
79 - Top 5 estudiantes con más ausencias
80 - Filtros AJAX (sin recarga):
81
82   - Asignatura
83   - Estudiante
84   - Estado
85   - Fecha desde / hasta
86 - Tabla completa de asistencias
87 - Corrección: estadísticas visibles aunque no haya resultados
88
89 Vista fallback: `no_tutor.php`
90 ---
91
92 ### 4■■■ Asignaciones Docentes Reorganizadas
93
94 ##### Docente-Materia
95 - Filtros avanzados
96 - Validación única por curso
97
98 ##### Tutor
99 - Vista independiente
100 - Selección inteligente de docentes elegibles
101 - Restricción: 1 curso por tutor
102 - Confirmaciones modales
103 - Dashboard muestra tutoría asignada
104
105 ---
106
107 ### 5■■■ Configuración Institucional Ampliada
108
109 Campos agregados:
110 - Provincia
111 - Ciudad
112 - Director
113 - Código AMIE
114 - Web
115 - Logo
116
117 Tabla nueva:
118 - `institution_shifts`
119
120 Funciones:
121 - Jornadas múltiples
122 - Select cascada Ecuador
123 - Autocompletar URL
124 - Gestión visual de jornadas
125
126 ---
127
128 ### 6■■■ Estructura Académica Ecuador
129
130 Soporte completo para:
131
132 - Inicial
133 - Educación General Básica
134 - Bachillerato General Unificado
135 - Bachillerato Técnico
136 - Figura profesional y especialidad
137 - Jornada nocturna condicionada
138 - Edición y creación de cursos con precarga automática
139
140 ---
141
142 ### 7■■■ Reportes PDF / Excel
143
144 - Datos institucionales dinámicos
145 - Vista previa estable
146 - Corrección de entidades HTML
147 - Eliminación de duplicidad de jornada
148 - Nombres de archivo sanitizados
149
150 ---
151
152 ### 8■■■ Módulo de Respaldos
153
154 - Interfaz completa

```



```
155 - Detección automática de `mysqldump`
156 - Validación de archivos
157 - Eliminación individual
158 - Limpieza automática
159
160 Ruta pendiente:

161 - `delete_backup`
162
163 ---
164
165 ### 9■■■ Gestión de Representantes
166
167 - Filtros dinámicos
168 - Eliminación con confirmación
169 - Inspector visualiza justificaciones revisadas
170
171 Ruta pendiente:
172 - `remove_representative`
173
174 ---
175
176 ### ■ Justificaciones
177
178 - Vista separada para revisadas
179 - Filtros por estado
180 - Métodos agregados en modelo y controlador
181
182 ---
183
184 ### 1■■■1■■■ Diseño Unificado Bootstrap
185
186 - `head.php` y `footer.php`
187 - Migración progresiva de vistas
188 - Dashboard completamente migrado
189 - Eliminación de navbar duplicado
190 - Estilos globales consistentes
191
192 ---
193
194 ### 1■■■2■■■ Navbar y Responsive Total
195
196 - Navbar reescrito completamente
197 - Botón hamburguesa en  $\leq 900\text{px}$ 
198 - Panel móvil desplegable interno
199 - Dropdown clic móvil / hover desktop
200 - Usuario y notificaciones accesibles en móvil
201 - Polling de notificaciones cada 30s
202 - Enlace directo: "Asistencia de Mi Curso"
203 - Dashboard adaptable:
204   - 3 columnas desktop
205   - 2 tablet
206   - 1 móvil
207
208 ---
209
210 ### 1■■■3■■■ UX/UI Global
211
212 - Navbar sticky
213 - Modales personalizados
214 - Filtros persistentes
215 - Badges visuales
216 - Orden lógico de menús
217 - Advertencias temporales
218 - Corrección de onclick por comillas
219 - Corrección de stats ocultos al filtrar
220
221 ---
222
223 ### 1■■■4■■■ Validaciones Críticas
224
225 - Asistencia sin duplicados
226 - Tutor único por curso
227 - Materia única por curso
228 - Horario sin conflictos
229 - Roles protegidos
230 - Estudiante único por año lectivo
231 - Jornadas múltiples controladas
232 - Eliminaciones protegidas
233
```

```
234 ---
235
236 ## ■■ BASE DE DATOS
237
238 ### Principales
239
240 institutions
241
242 users
243 roles
244 permissions
245
246 ### Académico
247
248 school_years
249 courses
250 subjects
251 teacher_assignments
252 course_students
253 class_schedule
254 institution_shifts
255
256
257 ### Asistencia
258
259 attendances
260 justifications
261
262
263 ### Sistema
264
265 notifications
266 activity_logs
267 representatives
268
269
270 ---
271
272 ## ■■ ARQUITECTURA DEL PROYECTO
273
274 MVC modular:
275
276 - Models especializados
277 - Controllers funcionales
278 - Helpers de seguridad, correo y respaldo
279 - Router central `public/index.php`
280 - BASE_PATH activo
281 - Views Bootstrap responsivas
282
283 ---
284
285 ## ■■ CONFIGURACIÓN
286
287 - Zona horaria Ecuador
288 - Sesiones persistentes 24h
289 - Timeout 30 min
290 - Cookies seguras
291 - Permisos en carpetas:
292
293 /uploads
294 /uploads/institution
295 /backups
296
297
298 Credencial prueba:
299
300 prof.diaz / password
301
302
303 URL:
304
305 http://localhost/EcuAsistencia2026/public/
306
307
308 ---
309
310 ## ■ BUGS CORREGIDOS
311
312 - Entidades HTML en nombres
```

```

313 - Jornadas duplicadas
314 - onclick roto
315 - Roles incorrectos
316 - Selectores sin filtrar
317 - Logo no guardado
318 - Sesiones mal inicializadas
319 - Vista previa de reportes
320 - Backups vacíos

321 - Falta de require_once en rutas
322 - Stats ocultos al no encontrar resultados
323
324 ---
325
326 ## ■ DEPENDENCIAS
327
328 phpmailer/phpmailer
329 phpooffice/phpspreadsheet
330 tecnickcom/tcpdf
331
332
333 ---
334
335 ## ■ PRÓXIMOS PASOS
336
337 ### Alta prioridad
338 1. Migrar vistas restantes a Bootstrap
339 2. Agregar rutas pendientes
340 3. Probar módulo backups completo
341 4. Validar rendimiento de consultas estadísticas
342
343 ### Media
344 5. Toast notifications
345 6. Breadcrumbs
346 7. Gráficos estadísticos
347
348 ### Baja
349 8. Modo oscuro
350 9. Calendario visual académico
351
352 ---
353
354 ## ■ ESTADO GENERAL
355
356 | Área          | Estado          |
357 |-----|-----|
358 | Arquitectura  | Estable         |
359 | Módulos       | Completos       |
360 | UX/UI         | Mejorada        |
361 | Responsive    | Implementado     |
362 | Seguridad     | Sólida          |
363 | Base datos    | Normalizada     |
364 | Optimización  | En progreso     |
365
366 ---
367
368 ## ■ CONCLUSIÓN
369
370 EcuAsist 2026 alcanzó madurez funcional con:
371
372 - Automatización académica sólida
373 - Sistema tutor avanzado con análisis dinámico
374 - Arquitectura modular escalable
375 - Consistencia visual responsiva
376 - Estabilidad operativa
377
378 Actualmente se encuentra en fase de optimización avanzada y preparación final para entorno productivo.
379
380 ---
381
382 **FIN DEL RESUMEN**

```

## 89. SugerenciasFuturas.md

Ruta: 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 > Asistencia > 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. REPORTE
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?

## 90. composer.json

Ruta: composer.json

```
1 {  
2     "require": {  
3         "phpmailer/phpmailer": "^7.0",  
4         "phpoffice/phpspreadsheet": "^5.4",  
5         "tecnickcom/tcpdf": "^6.10"  
6     }  
7 }  
8
```

## 91. malla.md

Ruta: 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

## 92. men#U00fa\_v2.0.md

Ruta: 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.