Documentación de Pruebas Unitarias y de Vistas

Proyecto: Módulos de Información, Analítica y Reportes, CADI, Eventos y Servicios

Autor: Generado por asistente • Fecha: 2025-10-19

# 1. Módulo de Información (Horarios personalizados y Notificaciones)

## 1.1 Modelos: HorarioPersonalizado

**Objetivo de la Prueba:** Validar creación, atributos y reglas de negocio de horarios personalizados.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| HorarioPersonalizado.objects.create | Crear horario válido (Clase de Programación) | usuario=estudiante; dia\_semana=1; hora\_inicio='08:00'; hora\_fin='10:00'; recordatorio=True | Horario creado; horas válidas; recuperable por título. |
| HorarioPersonalizado.objects.create | Validación de horas inválidas | hora\_inicio='16:00'; hora\_fin='14:00' | Lanza excepción por 'hora\_fin' < 'hora\_inicio'. |

## 1.2 Vistas: Horarios

**Objetivo de la Prueba:** Asegurar que las vistas de listar, crear, actualizar y eliminar horarios funcionan y respetan permisos.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| GET horarios:listar | Usuario ve solo sus horarios | login(estudiante); otros horarios de otro usuario existentes | Status 200; en contexto solo horarios del usuario. |
| POST horarios:crear | Crear un nuevo horario | titulo='Nuevo Horario'; dia\_semana=4; hora\_inicio='16:00'; hora\_fin='18:00' | 302 redirect; horario persistido y asociado al usuario actual. |
| POST horarios:actualizar | Actualizar horario propio | cambios en título, horas, color, recordatorio | 302; atributos actualizados. Si intenta actualizar ajeno → 403. |
| POST horarios:eliminar | Eliminar horario propio | pk horario | 302; objeto eliminado. Si intenta eliminar ajeno → 403. |
| POST horarios:sincronizar | Sincronizar actividad → horario | actividad\_id válido | 200; se crea HorarioPersonalizado relacionado a la actividad. |

## 1.3 Modelos & Vistas: Notificaciones

**Objetivo de la Prueba:** Validar preferencias, creación/marcado de notificaciones y listados.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| PreferenciaNotificacion.objects.create | Configurar preferencias por usuario | notificar\_actividades=True; notificar\_torneos=True; notificar\_psu=False; medio='Email' | Preferencias persistidas; usadas por lógica de envío. |
| Notificacion.objects.create / marcar\_como\_leida | Crear y marcar como leída | titulo='Recordatorio de actividad'; leida=False → marcar\_como\_leida() | Leída=True; conteos consistentes; recientes últimos 7 días. |
| utils.enviar\_notificacion | Respeta preferencias (PSU desactivado) | tipo='PSU' con preferencia desactivada | No se crea notificación nueva. |

# 2. Módulo de Analítica y Reportes

## 2.1 Modelos

**Objetivo de la Prueba:** Verificar registros estadísticos, reportes, permisos de exportación, control de asistencia y auditoría de accesos.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| DatoEstadistico.objects.create | Crear datos de participación | categoria='Participación'; subcategoria='Deportes'; periodo='2025-Q3'; cantidad=150; porcentaje=15.5 | Registros creados y filtrables por categoría; suma total consistente. |
| Reporte.objects.create | Generar reporte con datos vinculados | titulo='Reporte Trimestral'; tipo='Participación'; datos\_incluidos='dato1,dato2' | Reporte persistido; relación a datos válida. |
| PermisoExportacion.objects.create / tiene\_permiso | Permisos por usuario/rol | profesor con CSV/Excel; admin implícitamente permitido | Validación True/False según reglas. |
| ControlAsistencia.objects.create | Registrar asistencias | actividad, estudiante, presente=True, metodo='QR' | Registros consultables por estudiante; conteos correctos. |
| RegistroAcceso.objects.create | Auditar acciones sobre reportes | accion='Consultar Reporte'/'Exportar Reporte' | Registros filtrables y ordenables por fecha. |

## 2.2 Vistas

**Objetivo de la Prueba:** Validar acceso al dashboard, generación/exportación/visualización de reportes e importación CSV.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| GET reportes:dashboard | Accesos por rol | admin y profesores → acceso; estudiante → 403 | 200 para admin/profesor; 403 estudiante; datos en contexto. |
| POST reportes:generar | Generar nuevo reporte | titulo, tipo, periodo, filtros JSON | 200; creado; RegistroAcceso='Generar Reporte'. Estudiante → 403. |
| GET reportes:exportar?formato=csv | Exportación CSV por profesor | pk reporte | 200; Content-Type text/csv; archivo descargable; RegistroAcceso='Exportar Reporte'. Estudiante → 403. Formato sin permiso → 403. |
| GET reportes:detalle | Visualizar reporte | pk reporte | 200; registro de acceso 'Consultar Reporte'. |
| POST datos:importar (CSV) | Importación de datos estadísticos desde CSV | archivo CSV válido / inválido | 200; crea registros; valida tipos y rechaza datos inválidos. |

# 3. Módulo CADI (Actividades)

## 3.1 Modelos

**Objetivo de la Prueba:** Comprobar tipos, lugares, actividades y valoraciones.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| TipoActividad.objects.create | Crear tipos de actividad | nombre='Deporte'/'Arte' | Tipos persistidos; recuperables por nombre. |
| Lugar.objects.create | Crear lugares con capacidad | nombre='Cancha Múltiple'; capacidad=30 | Lugar creado; capacidad correcta. |
| Actividad.objects.create | Crear actividad válida | fechas inicio<fin; cupos\_totales=20; instructor | Actividad creada; cupos\_disponibles inicial = cupos\_totales; inscribir reduce cupos. |
| Valoracion.objects.create | Valorar actividad | puntuacion=4; comentario | Valoración creada; restricción: 1 por usuario/actividad. |

## 3.2 Vistas

**Objetivo de la Prueba:** Validar listados, filtros y creación/actualización de actividades (según rol).

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| GET actividades:listar | Listado básico | login estudiante | 200; actividades en contexto. |
| GET actividades:listar?tipo=<id> | Filtrar por tipo | tipo existente / sin actividades | 200; lista filtrada; posible 0 resultados. |
| GET actividades:listar?lugar=<id> | Filtrar por lugar | lugar existente | 200; lista filtrada. |
| GET actividades:listar?fecha=YYYY-MM-DD | Filtrar por fecha | fecha con/sin actividades | 200; lista filtrada; posible 0 resultados. |
| POST actividades:crear | Crear actividad (admin) | datos completos | 302; creada; usuario no admin → 403. |
| POST actividades:actualizar | Actualizar actividad (admin) | datos actualizados | 302; cambios persistidos. |

# 4. Módulo de Eventos y Servicios (Torneos, PSU, Citas psicológicas)

## 4.1 Torneos

**Objetivo de la Prueba:** Validar ciclo de vida de torneos, equipos y partidos.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| Torneo.objects.create | Crear torneo válido | tipo\_inscripcion='Equipo'; fecha\_inicio < fecha\_fin | Torneo creado; validación de fechas activa. |
| Equipo.objects.create / EquipoMiembro.objects.create | Crear equipo y miembros (límites) | max\_participantes\_equipo=5 | Miembros hasta el máximo; exceder → excepción. |
| Partido/Resultado.objects.create | Programar y registrar resultado | puntos\_local=3; puntos\_visitante=1 | Resultado registrado; partido pasa a 'Finalizado'. |
| GET torneos:listar / detalle | Visibilidad de torneos | anónimo o autenticado | 200; detalle con campos del torneo. |
| POST torneos:crear (admin) | Crear torneo (permisos) | datos completos | 302; estudiante → 403. |
| POST torneos:inscribir\_equipo | Inscripción de equipo por estudiante | torneo en 'Inscripciones'; nombre equipo; miembros | 200; equipo creado; doble inscripción → 400; si torneo≠'Inscripciones' → 400. |
| POST torneos:programar\_partido (admin) | Programar partido | equipos existentes; fecha; lugar; ronda | 200; partido creado; no admin → 403. |
| POST torneos:registrar\_resultado (admin) | Registrar resultado | partido\_id; puntos; comentarios | 200; estado partido 'Finalizado'; no admin → 403. |

## 4.2 Proyecto Social Universitario (PSU)

**Objetivo de la Prueba:** Validar alta de proyectos, inscripción y registro de horas.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| ProyectoSocial.objects.create | Crear proyectos válidos | fechas inicio ≤ fin; cupos y cupos\_disponibles iguales al crear | Proyecto creado; validación de fechas. |
| InscripcionPSU.objects.create | Inscribir estudiante | proyecto activo con cupos>0 | Inscripción creada; decremento de cupos; duplicado → excepción; sin cupos → excepción. |
| POST psu:registrar\_horas | Registrar horas acumuladas | horas 10 + 30 (total 40) | Estado cambia a 'Completado'; estudiantes sin permisos → 403. |

## 4.3 Citas Psicológicas

**Objetivo de la Prueba:** Validar creación, reserva, cancelación y vistas de citas.

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Valores de Entrada | Resultado esperado |
| CitaPsicologica.objects.create | Crear slots disponibles | psicólogo; fecha\_hora; duracion=60; estado='Disponible' | Citas disponibles persistidas. |
| cita.reservar / cancelar | Reservar y cancelar cita | estudiante reserva; reintento de reservar ocupada | Asignación estudiante; estado 'Reservada'; reservar de nuevo → excepción; cancelar → 'Disponible'. |
| Vistas (listar/reservar/cancelar) | Flujo completo vía vistas | endpoints de citas (según implementación) | 200 en operaciones válidas; validaciones y permisos aplicados. |

Nota: Esta documentación fue generada a partir de los archivos de pruebas del proyecto.