

Análisis de pruebas unitarias:

Las pruebas de nuestro proyecto se pueden resumir de esta forma para todas las clases/tipos de los tests (analizando brevemente cada clase para cada archivo de tests):

Los tests de autenticación de usuarios se dividen en dos clases de esta manera:

1. AcademicMembersLoginTestCase:

Evalúan la funcionalidad de inicio de sesión para miembros académicos en la aplicación web. Comprueban la respuesta a las solicitudes GET y POST, verifican la autenticación con credenciales válidas e inválidas y aseguran que los usuarios pertenezcan al grupo adecuado.

2. AuthViewsTestCase:

Abarcan la autenticación y cierre de sesión en diferentes vistas de la aplicación. Examinan tanto las solicitudes GET como POST para las vistas de inicio de sesión, validan las credenciales del usuario y garantizan que la sesión se cierre correctamente, redirigiendo al usuario a la página de inicio.

Los tests para el cortex de la página se pueden definir como:

1. CortexViewTestCase:

Comprenden pruebas para la vista principal ('home'), la vista index para diferentes grupos de usuarios, y aseguran que las respuestas sean exitosas y que se utilicen las plantillas adecuadas según el contexto de autenticación del usuario y su pertenencia a grupos específicos.

Los tests para los eventos, solicitudes y ceremonias tienen muchas diferencias y bifurcaciones, pero de forma simple los podemos poner de esta forma:

1. CreateEventRequestTestCase:

Evalúan la creación de solicitudes de eventos. Verifican el acceso a la vista, la correcta visualización del formulario, el envío correcto del formulario con los datos adecuados, y la ocultación del campo de profesor para ciertos grupos de usuarios.

2. EventRequestRecordTestCase:

Verifican el acceso a la vista de registro de solicitudes de evento para usuarios autenticados, filtrado de resultados para diferentes grupos, y búsqueda por título.

3. EventRequestListTestCase:

Evalúan la lista de solicitudes del evento. Se prueba el acceso a la vista, la visualización correcta de eventos, y la capacidad de filtrar eventos por título.

4. EventListTestCase:

Verifican la lista de eventos. Se evalúa el acceso tanto para usuarios autenticados como no autenticados, la correcta visualización de eventos, y la capacidad de búsqueda y filtrado por título y estado.

5. SaveTasksTestCase:

Evalúan la capacidad de guardar tareas relacionadas con eventos, como lugar, presupuesto, alimentación y transporte.

6. EventRegistryTestCase:

Verifican el registro de eventos. Se evalúa el acceso a la vista, la capacidad de búsqueda y filtrado por ID y título, y la correcta visualización de eventos finalizados.

7. ResetCeremonyTestCase:

Evalúan la funcionalidad de restablecimiento de ceremonia, incluyendo la capacidad de restablecer actividades de ceremonia y la redirección adecuada después de restablecer.

8. GuardarEventoTestCase:

Verifican la capacidad de guardar eventos con diferentes estados para alimentación, transporte y extras.

9. EventListApoyoTestCase:

Pruebas para verificar que solo se muestran los eventos en curso en la lista de apoyo.

10. FinishEventApoyoTestCase:

Pruebas para verificar que solo se muestran los eventos finalizados en la lista de apoyo.

Los tests para formularios se basan en la información que se proporciona en las solicitudes y su gestión, y ceremonias de esta forma:

1. TestEventRequestForm:

Se centran en el formulario de solicitud de eventos. Verifica que el formulario identifique correctamente los campos faltantes, maneje los rangos de fecha inválidos y los presupuestos negativos. Además, comprueba que el formulario acepte datos válidos y que pueda crear, editar y eliminar solicitudes de eventos, así como también crear solicitudes de estado.

2. TestEstadoSolicitudForm:

Se enfocan en el formulario de estado de solicitud. Asegura que el formulario acepte datos válidos para crear solicitudes de estado asociadas a eventos.

3. TestCeremonyActivityForm:

Validan el formulario para crear actividades de ceremonia. Se asegura de que el formulario acepte datos válidos para crear actividades de ceremonia.

Los tests de los modelos se basan en la información de la base de datos y como se puede probar su gestión:

1. TestCreateModels:

Crea las instancias para varios modelos de la aplicación. Verifica que las instancias creadas existan en la base de datos después de la creación.

2. TestEditModels:

Validan la capacidad de editar instancias de los modelos existentes. Se aseguran de que los cambios realizados en las instancias se reflejen correctamente en la base de datos.

3. TestDeleteModels:

Se enfocan en la eliminación de instancias de modelos. Asegura que las instancias eliminadas se eliminen correctamente de la base de datos y que el recuento de instancias sea cero después de la eliminación.

Finalmente, los tests para los usuarios y sus funcionalidades se definen de esta forma:

1. **UserTestCase:**

Suite de pruebas que evalúa funcionalidades relacionadas con usuarios, como la creación de usuarios comunes y super usuarios, así como la asignación de roles de grupo.

Ejecución y análisis de la cobertura de los tests:

Name	Stmts	Miss	Cover
manage.py	12	2	83%
myapp__init__.py	0	0	100%
myapp\admin.py	3	0	100%
myapp\apps.py	4	0	100%
myapp\context_processors.py	4	0	100%
myapp\forms.py	28	0	100%
myapp\migrations\0001_initial.py	7	0	100%
myapp\migrations\0002_notification_url.py	4	0	100%
myapp\migrations\0003_remove_ceremonyactivity_ceremony_and_more.py	5	0	100%
myapp\migrations\0004_remove_ceremony_ceremony_activities_and_more.py	5	0	100%
myapp\migrations\0005_alter_ceremonyactivity_ceremony.py	5	0	100%
myapp\migrations\0006_event_descripcion_event_titulo_and_more.py	4	0	100%
myapp\migrations\0007_event_estado_alimentacion_event_estado_extras_and_more.py	4	0	100%
myapp\migrations__init__.py	0	0	100%
myapp\models.py	57	4	93%
myapp\urls.py	6	0	100%
myapp\views__init__.py	4	0	100%
myapp\views\auth.py	29	1	97%
myapp\views\cortex.py	24	1	96%
myapp\views\event.py	147	41	72%
myapp\views\eventRequest.py	99	28	72%
mysite__init__.py	0	0	100%
mysite\settings.py	37	3	92%
mysite\urls.py	3	0	100%
test\unit\test_auth.py	109	2	98%
test\unit\test_cortex.py	43	0	100%
test\unit\test_event.py	242	16	93%
test\unit\test_forms.py	45	0	100%
test\unit\test_models.py	85	8	91%
test\unit\test_user.py	28	11	61%
TOTAL	1043	117	89%

De forma general, se puede apreciar que la cobertura es bastante buena, teniendo un promedio de un 89% de cobertura sobre el código.

Además, adentrándonos en este promedio, se puede apreciar que si bien la mayoría de archivos con cobertura de un 100% son migraciones y archivos secundarios a los tests principales, los principales no se quedan tan atrás en su mayoría, y ahondan entre un 70% y un 90% de cobertura, lo cual sigue siendo bueno.

Ejecución y cobertura de los tests con omisiones:

Ahora, omitiendo estos archivos secundarios, la cobertura se puede ver así:

Name	Stmts	Miss	Cover

myapp\context_processors.py	4	0	100%
myapp\forms.py	28	0	100%
myapp\models.py	57	4	93%
myapp\views__init__.py	4	0	100%
myapp\views\auth.py	29	1	97%
myapp\views\cortex.py	24	1	96%
myapp\views\event.py	147	41	72%
myapp\views\eventRequest.py	99	28	72%

TOTAL	392	75	81%

Como se puede observar junto al informe anterior, al omitir archivos como settings.py, migrations, manage.py, admin.py, etc, la cobertura en promedio se redujo a de un 89% a un 81%, lo cual sigue siendo bastante bien para las pruebas de estos archivos que son vitales para el proyecto.

Si bien la cobertura de algunos archivos se ve comprometida y se reduce ligeramente, no afecta de manera general a la cobertura del proyecto, permaneciendo incluso a pesar de esto por encima del 80% de cobertura, siendo lo óptimo y necesario para la fidelidad del proyecto y su funcionamiento correcto.

Ejecución individual de las pruebas E2E:

```
Found 13 test(s).  
Creating test database for alias 'default'...  
System check identified no issues (0 silenced).
```

```
DevTools listening on ws://127.0.0.1:58419/devtools/browser/1ab4023e-7ffc-4b1f-9950-be29b9844c85  
.
```

```
DevTools listening on ws://127.0.0.1:58514/devtools/browser/b3a04870-4cc2-4ac0-b6d2-862f759c1d4d  
.
```

```
DevTools listening on ws://127.0.0.1:58610/devtools/browser/1ad7dffc-07ad-4393-95bd-f944577e8df9  
.
```

```
DevTools listening on ws://127.0.0.1:58716/devtools/browser/b1ca2477-1eb0-4003-b963-8a153e61da17  
.
```

```
DevTools listening on ws://127.0.0.1:58805/devtools/browser/7c58ce25-6d97-4dcd-acbb-6d614a782c54  
.
```

```
DevTools listening on ws://127.0.0.1:58879/devtools/browser/3ad533b1-f1a2-48f6-84d1-8aa717a805f0  
.
```

```
DevTools listening on ws://127.0.0.1:58966/devtools/browser/99d237fc-eb15-4f1e-b0de-7d599338ece4  
.
```

```
DevTools listening on ws://127.0.0.1:59053/devtools/browser/75539391-5b8e-4a08-8292-c8c6f0eeb691  
.
```

```
DevTools listening on ws://127.0.0.1:59152/devtools/browser/447ad4ae-7636-483b-a8dd-ac1616710ad4  
.
```

```
DevTools listening on ws://127.0.0.1:59268/devtools/browser/138c106a-96b4-49a0-a898-78b195f9fdd2  
.
```

```
DevTools listening on ws://127.0.0.1:59364/devtools/browser/8b996382-4fab-4919-aadd-99476c297d2e  
Ninguno de los campos puede estar vacío.  
.
```

```
DevTools listening on ws://127.0.0.1:59458/devtools/browser/95e78084-72a4-45b7-b8f4-fc0c6c18c934  
.
```

```
DevTools listening on ws://127.0.0.1:59561/devtools/browser/73d393c2-7e9c-4d48-8614-f917d8a6b89b
.
-----
Ran 13 tests in 266.800s
```

Cálculo de indicadores:

- Número de pruebas ejecutadas: 13
- Número de incidentes por acción en página: 0
- Número de casos de prueba re-ejecutados: 0
- Tiempo Promedio de Resolución de Defectos: 0 seg
- Completitud de pruebas:

$$\text{Completitud de pruebas} = \frac{\text{Pruebas ejecutadas}}{\text{Pruebas planificadas}} \times 100\%$$

$$\text{Completitud de pruebas} = \frac{13}{30} \times 100\% = 0.43 \text{ de completitud}$$

- Densidad de errores:

$$\text{Densidad de errores} = \frac{\# \text{ promedio de defectos}}{\text{KLOC}}$$

Donde:

1. KLOC representa las líneas de código por miles.
2. Una falla por cada 1000 líneas (LOC) se considera aceptable según las mejores prácticas

$$\text{Densidad de errores} = \frac{0}{1354 \text{ lineas}} = 0$$