

Gestión de Entradas para Micro-Eventos

Integrantes: Carlos Arévalo
Daniel Zeballos

Repositorio GitHub: <https://github.com/ProductoraEventos-Proyecto/gestion-eventos-app>

1. Especificación de Requerimiento

El requerimiento inicial del cliente era incompleto, por lo que realizamos un proceso de validación para asegurar que el sistema cumpla con las expectativas del cliente. Se han definido los criterios de aceptación de cada funcionalidad:

- **CRUD Evento:**
 - Crear Evento: Nombre, fecha, categoría, precio de entrada y cupos disponibles serán campos obligatorios. Los campos no deben estar vacíos; precio y cupos deben ser números enteros y el precio debe ser positivo.
 - Editar Evento: Permite modificar cualquier campo de un evento creado por sí mismo. Las validaciones son las mismas que en la creación.
 - Eliminar Evento: Se permite eliminar cualquier evento propio.
 - Consultar Evento: Mostrar listado completo de eventos con información clara de cada campo. El listado deberá estar siempre actualizado frente a acciones CRUD del sistema.
- **Filtrado y Búsqueda de Eventos:** se puede filtrar por nombre, descripción, categoría, fecha y un rango de precio. Los resultados deben coincidir exactamente con todos los criterios ingresados.
- **Generación de Reportes:** Debe mostrar el número de eventos registrados, el número de cupos disponibles para cada uno de los eventos y el número de eventos con cantidad de cupos menores a 0.
- **Autenticación y Creación de Usuarios:** El usuario y contraseña serán obligatorios en ambos casos. No se pedirán requisitos mínimos para ninguna de las dos entradas. Al equivocarse logueando se enviará un mensaje de error que no especifica cuál de las dos entradas es incorrecta. La validación de los usuarios será sensible a mayúsculas/minúsculas y caracteres especiales.

2. Cumplimiento de Requerimientos

Para realizar la verificación del sistema se realizarán pruebas unitarias al código por medio de pytest a las principales funcionalidades del sistema. Además se contrastaran los requerimientos y las pruebas de tal forma que nos aseguremos que la aplicación cumple correctamente cada uno de los requerimientos planteados.

3. Organización y Flujo de Trabajo

3.1. Organización:

Cada integrante está a cargo de realizar 2 módulos de la aplicación. Carlos Arevalo desarrollará el módulo de autenticación y el módulo CRUD de

eventos. Daniel Zeballo desarrollará el módulo de búsqueda y filtrado y el módulo de registro de eventos.

3.2. Flujo de trabajo

Se trabajara con un repositorio único en [GitHub](#) el cual se organizara cómo un GitFlow, por lo que se habilitó protección en main para evitar commits directos.

Para el manejo de logs se habilitará [Sentry.io](#). Para dejar constancia de las pruebas se utilizará [Greentest.ai](#). Se creará un espacio de trabajo del equipo en [Slack.com](#) el cual estará conectado tanto con GitHub cómo con Sentry.io.

4. Problemas y Soluciones

Durante la realización de la tarea se tuvieron varios problemas, cómo es el hecho de que inicialmente no habíamos entendido por completo cómo usar el flujo de trabajo de GitFlow por lo cual algunas cosas se subieron de forma desordenada, sin embargo investigamos y comprendimos cómo se tenía que hacer, por lo que arreglamos las ramas y trabajamos de forma buena y ordenada.

Otro problema que se tuvo fue que algunos testing que tenían que ver con tkinter daban error porque la interfaz gráfica no está diseñada para ejecutarse en entornos de pruebas automatizadas, entonces se hizo con una estrategia de mocking que es simular la interacción de con los componentes de la GUI.

5. Pruebas Realizadas

Para el testing de la aplicación se van a realizar pruebas unitarias a distintos formularios de la aplicación, inicialmente cada integrante realizará pruebas a ciertos formularios, para luego unificar las pruebas realizadas. Carlos Arévalo se encargará de realizar las pruebas a los formularios que afectan directamente a los eventos. Daniel Zeballos se encargará de las pruebas realizadas al login y el registro de usuarios de la aplicación. A continuación se deja constancia de las pruebas realizadas luego de la unificación.

Id_Test	Entrada al sistema	Resultado Esperado	Resultado Obtenido	Fallo o Éxito	Comentario adicional
test_iniciar_sesion_exitoso	Usuario: test_user, Contraseña: password123	Se inicia sesión correctamente y se llama a on_success_callback("test_user").	Se inicia sesión correctamente y se llama a on_success_callback("test_user").	Éxito	Inicio de sesión correcto con username y password válidos.
test_iniciar_sesion_incorrecto	Usuario: test_user, Contraseña: wrong_pass	Se muestra error "Usuario o contraseña incorrectos".	Se muestra error "Usuario o contraseña incorrectos".	Éxito	Inicio de sesión falla correctamente con password incorrecto.

test_iniciar_sesion_ambos_vacios_username	Usuario vacío, Contraseña: pass	Se muestra error “Usuario y contraseña no pueden estar vacíos”.	Verificar_usuario fue llamado aun cuando username estaba vacío.	Fallo	
test_iniciar_sesion_ambos_vacios_password	Usuario: user, Contraseña vacío	Se muestra error “Usuario y contraseña no pueden estar vacíos”.	Verificar_usuario fue llamado aun cuando password estaba vacío.	Fallo	
test_iniciar_sesion_ambos_vacios_total	Usuario vacío, Contraseña vacío	Se muestra error “Usuario y contraseña no pueden estar vacíos”.	Verificar_usuario fue llamado aun cuando ambos campos estaban vacíos.	Fallo	
test_iniciar_sesion_username_con_espacios	Usuario con espacios "user123", Contraseña: password	Se eliminan espacios y se valida usuario "user123".	Username no se recorta antes de verificar.	Fallo	
test_iniciar_sesion_password_con_espacios	Usuario: user, Contraseña con espacios "pass "	El sistema valida con espacios tal cual, debe fallar autenticación.	El sistema valida con espacios tal cual, debe fallar autenticación.	Éxito	Password con espacios produce error como se esperaba.
test_iniciar_sesion_sensible_a_mayusculas	Usuario: User, Contraseña: pass	Diferencia de mayúsculas/minúsculas → credenciales inválidas.	Diferencia de mayúsculas/minúsculas → credenciales inválidas.	Éxito	Login sensible a mayúsculas falla correctamente.
test_iniciar_sesion_username_unicode	Usuario: usuario ñ, Contraseña: clave	Se autentica correctamente con caracteres Unicode.	Se autentica correctamente con caracteres Unicode.	Éxito	Username con caracteres unicode se procesa correctamente.
test_iniciar_sesion_username_caracteres_especiales	Usuario: user!@#, Contraseña: pass!@#	Se autentica correctamente con caracteres especiales.	Se autentica correctamente con caracteres especiales.	Éxito	Username con caracteres especiales se procesa correctamente.
test_iniciar_sesion_username_largo	Usuario largo (200 caracteres), Contraseña: password	Se procesa y valida el usuario largo correctamente.	Se procesa y valida el usuario largo correctamente.	Éxito	Username largo se procesa correctamente.

test_iniciar_sesion_password_largo	Usuario: user, Contraseña larga (200 caracteres)	Se procesa y valida la contraseña larga correctamente.	Se procesa y valida la contraseña larga correctamente.	Éxito	Password largo se procesa correctamente.
test_crear_cuenta_exitoso	Usuario nuevo: new_user, Contraseña: new_pass	Se crea cuenta nueva exitosamente y se muestra confirmación.	Se crea cuenta nueva exitosamente y se muestra confirmación.	Éxito	Cuenta creada exitosamente.
test_crear_cuenta_existente	Usuario existente: existente, Contraseña: pass	Se muestra error "El usuario ya existe. Por favor, elige otro".	Se muestra error "El usuario ya existe. Por favor, elige otro".	Éxito	Intento de crear cuenta existente falla correctamente.
test_crear_cuenta_campos_vacios	Usuario vacío, Contraseña vacío	Se muestra error "Usuario y contraseña no pueden estar vacíos".	Se muestra error "Usuario y contraseña no pueden estar vacíos".	Éxito	Validación de campos vacíos funciona correctamente.
test_crear_cuenta_username_con_espacios	Usuario con espacios " user ", Contraseña: pass	Se eliminan espacios y se crea usuario "user".	Username no se recorta antes de crear cuenta.	Fallo	
test_crear_cuenta_password_con_espacios	Usuario: user, Contraseña con espacios " pass "	Se crea usuario tal cual con espacios en contraseña.	Se crea usuario tal cual con espacios en contraseña.	Éxito	Password con espacios se procesa correctamente.
test_crear_cuenta_username_unicode	Usuario Unicode "软件测试", Contraseña: clave	Se crea cuenta correctamente con Unicode.	Se crea cuenta correctamente con Unicode.	Éxito	Username con caracteres unicode se procesa correctamente.
test_crear_cuenta_username_largo	Usuario largo (300 caracteres), Contraseña: pass	Se crea cuenta exitosamente con nombre largo.	Se crea cuenta exitosamente con nombre largo.	Éxito	Username largo se procesa correctamente.

test_crear_cuenta_password_largo	Usuario: user, Contraseña larga (300 caracteres)	Se crea cuenta exitosamente con contraseña larga.	Se crea cuenta exitosamente con contraseña larga.	Éxito	Password largo se procesa correctamente.
test_crear_cuenta_username_igual_a_password	Usuario = Contraseña = "same"	Se crea cuenta exitosamente aunque usuario = contraseña.	Se crea cuenta exitosamente aunque usuario = contraseña.	Éxito	Username igual a password se permite y se procesa correctamente.
test_volver_a_login_restaurar_estado	Navegar a crear cuenta y volver al login	Modo vuelve a "login" y subtítulo cambia a "Inicia sesión o crea tu cuenta".	Modo vuelve a "login" y subtítulo cambia a "Inicia sesión o crea tu cuenta".	Éxito	Volver a login restaura correctamente el estado de la UI.
test_crear_cuenta_cambiar_estado	Entrar a crear cuenta	Modo cambia a "register" y subtítulo cambia a "Creando nueva cuenta".	Modo cambia a "register" y subtítulo cambia a "Creando nueva cuenta".	Éxito	Crear cuenta cambia correctamente el estado de la UI.
test_volver_a_login_limpiar_campos	Ingresar datos, luego volver al login	Los campos de usuario y contraseña se vacían.	Los campos de usuario y contraseña se vacían.	Éxito	Volver a login limpia correctamente los campos de entrada.
test_crear_cuenta_ocultar_botón_login	Cambiar a crear cuenta	El botón de login desaparece.	El botón de login desaparece.	Éxito	Crear cuenta oculta correctamente el botón de login.
test_crear_cuenta_mostrar_botón_volver	Cambiar a crear cuenta	El botón "volver" se hace visible.	El botón "volver" se hace visible.	Éxito	Crear cuenta muestra correctamente el botón de volver.
test_crear_eventos_varios_1	"Concierto", "Música en vivo", "10-09-2025", "Musica", "5000", "100"	Se llama a crear_evento	Se llama a crear_evento	Éxito	Evento básico creado correctamente.
test_crear_eventos_varios_2	nombre_entry:"E*500"	Se llama a crear_evento	Se llama a crear_evento	Éxito	Evento con nombre muy largo se crea.
test_crear_eventos_varios_3	descripcion_entry:""	Se llama a crear_evento	Se llama a crear_evento	Éxito	

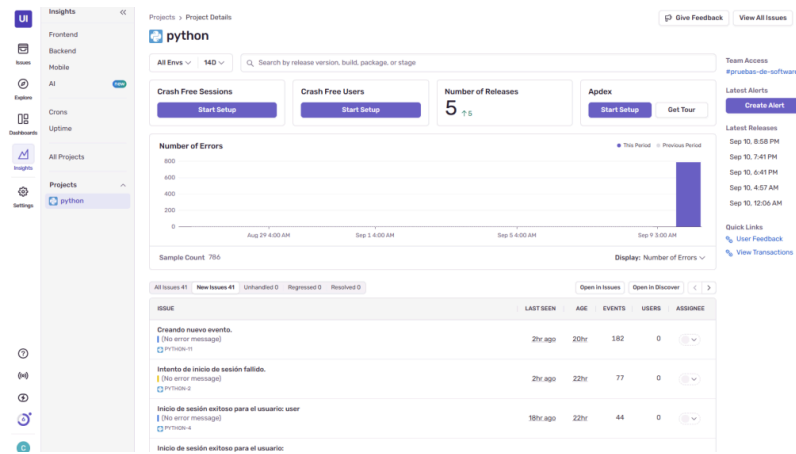
test_crear_eventos_varios_4	nombre_entry:""	Nos se llama a crear_evento	Nos se llama a crear_evento	Éxito	Evento sin nombre rechazado
test_crear_eventos_varios_5	fecha_entry:"32-13-2025"	No se llama a crear_evento	No se llama a crear_evento	Éxito	Evento con fecha inválida no se crea
test_crear_eventos_varios_6	precio_entry:"abc"	No se llama a crear_evento	No se llama a crear_evento	Éxito	Precio inválido detectado, evento no creado.
test_crear_eventos_varios_7	cupos_entry:"abc"	No se llama a crear_evento	No se llama a crear_evento	Éxito	Cupos inválidos detectados, evento no creado.
test_crear_eventos_varios_8	nombre_entry:"🎉Evento🎵", descripcion_entry:"🎤🎶🌟"	Se llama a crear_evento	Se llama a crear_evento	Éxito	Evento con emojis se crea correctamente.
test_crear_eventos_varios_9	precio_entry:"-500"	No se llama a crear_evento	No se llama a crear_evento	Éxito	Evento con precio negativo no aceptado
test_crear_eventos_varios_10	cupos_entry:"0"	No se llama a crear_evento	No se llama a crear_evento	Éxito	Evento con cupos cero no se crea
test_actualizar_evento_valido	ID seleccionado = 1, todos los campos llenos con datos válidos	Se llama a actualizar_evento	Se llama a actualizar_evento	Éxito	Evento actualizado exitosamente.
test_actualizar_evento_nombre_vacio	ID seleccionado = 1, campo nombre vacío	No se llama a actualizar_evento	No se llama a actualizar_evento	Éxito	Actualización rechazada por nombre vacío.
test_actualizar_evento_unicode	ID seleccionado = 2, campos con caracteres Unicode	Se llama a actualizar_evento	Se llama a actualizar_evento	Éxito	Se crea con éxito el unicode
test_eliminar_evento_propietario	Selección de evento propio en la lista	Se llama a eliminar_evento, muestra mensaje info	Se llama a eliminar_evento, muestra mensaje info	Éxito	Usuario puede eliminar su propio evento.
test_eliminar_evento_ajeno	Selección de evento de otro usuario	No se llama a eliminar_evento	No se llama a eliminar_evento	Éxito	Usuario no puede eliminar evento ajeno.

test_eliminar_evento_sin_seleccion	Ningún evento seleccionado	No se llama a eliminar_evento	No se llama a eliminar_evento	Éxito	Eliminación ignorada al no seleccionar evento.
test_búsqueda_nombre	Texto de búsqueda “Concierto” en campo nombre	Se llama a actualizar_listas_eventos	Se llama a actualizar_listas_eventos	Éxito	Muestra los elementos relacionados a la búsqueda.
test_búsqueda_categoria	Texto de búsqueda “Musica” en categoría	Se llama a actualizar_listas_eventos	Se llama a actualizar_listas_eventos	Éxito	Muestra los eventos relacionados con la categoría.
test_búsqueda_precio_min_max	Texto de búsqueda precio mínimo = 1000, máximo = 5000	Se llama a actualizar_listas_eventos	Se llama a actualizar_listas_eventos	Éxito	Filtra a travez de precio minimo y maximo
test_búsqueda_fecha	Texto de búsqueda fecha = 10-09-2025	Se llama a actualizar_listas_eventos	Se llama a actualizar_listas_eventos	Éxito	Busca los elementos de la fecha
test_mostrar_formulario	Llamar a mostrar_formulario()	El frame de formulario se muestra	El frame de formulario se muestra	Éxito	El formulario se visualiza correctamente
test_mostrar_registros	Llamar a mostrar_registros()	El frame de registros se muestra	El frame de registros se muestra	Éxito	test_mostrar_registros
test_habilitar_campos	Llamar a habilitar_campos()	Los campos quedan en estado “normal”	Los campos quedan en estado “normal”	Éxito	Los campos de entrada se habilitan correctamente.
test_deshabilitar_campos	Llamar a deshabilitar_campos()	Los campos quedan en estado “disabled”	Los campos quedan en estado “disabled”	Éxito	Los campos de entrada se deshabilitan correctamente.
test_limpiar_campos_sin_resetear_id	Campo nombre con texto, ID seleccionado = 99	Campos vacíos, ID no cambia	Campos vacíos, ID no cambia	Éxito	Los campos se limpian sin modificar el ID de evento seleccionado.
test_nombre_un_caracter	Nombre = “A”, demás campos válidos	Se llama a crear_evento	Se llama a crear_evento	Éxito	El sistema permite nombres con 1 caracter
test_descripción_larga	Descripción = 1000 caracteres, demás campos válidos	Se llama a crear_evento	Se llama a crear_evento	Éxito	Crea evento con descripcion larga

test_precio_grande	Precio = 1000000000, demás campos válidos	Se llama a crear_evento	Se llama a crear_evento	Éxito	Crea evento con precio elevado
test_cupos_grande	Cupos = 100000, demás campos válidos	Se llama a crear_evento	Se llama a crear_evento	Éxito	Crea evento con cupo elevado

6. Evidencia

6.1. Manejo de Logs en [Sentry.io](#)



6.2. Realización de pruebas y respaldo en [Greentest.ai](#)

ID	Nombre	Ciclos	Fig 1	Última ejecución
13	test_iniciar_sesion_exitoso	1	✓	✓
14	test_iniciar_sesion_incorrecto	1	✓	✓
15	test_iniciar_sesion_campos_vacios_username	1	✗	✗
16	test_iniciar_sesion_campos_vacios_password	1	✗	✗
17	test_iniciar_sesion_campos_vacios_total	1	✗	✗
18	test_iniciar_sesion_username_con_espacios	1	✗	✗
19	test_iniciar_sesion_password_con_espacios	1	✓	✓
20	test_iniciar_sesion_sensible_a_mayusculas	1	✓	✓
21	test_iniciar_sesion_username_unicode	1	✓	✓
22	test_iniciar_sesion_username_caracteres_especiales	1	✓	✓
23	test_iniciar_sesion_username_largo	1	✓	✓
24	test_iniciar_sesion_password_largo	1	✓	✓
25	test_crear_cuenta_exitoso	1	✓	✓
26	test_crear_cuenta_existente	1	✓	✓
27	test_crear_cuenta_campos_vacios	1	✓	✓
28	test_crear_cuenta_username_con_espacios	1	✗	✗
29	test_crear_cuenta_password_con_espacios	1	✓	✓
30	test_crear_cuenta_username_unicode	1	✓	✓
31	test_crear_cuenta_username_largo	1	✓	✓
32	test_crear_cuenta_password_largo	1	✓	✓

<input type="checkbox"/>	ID	Nombre	Ciclos	Obj	Última ejecución
<input type="checkbox"/>	1	test_crear_eventos_varios_1	10	✓	✓
<input type="checkbox"/>	2	test_crear_eventos_varios_2	10	✓	✓
<input type="checkbox"/>	3	test_crear_eventos_varios_3	10	✓	✓
<input type="checkbox"/>	4	test_crear_eventos_varios_4	10	✓	✓
<input type="checkbox"/>	5	test_crear_eventos_varios_5	10	✓	✓
<input type="checkbox"/>	6	test_crear_eventos_varios_6	10	✓	✓
<input type="checkbox"/>	8	test_crear_eventos_varios_7	10	✓	✓
<input type="checkbox"/>	9	test_crear_eventos_varios_8	10	✓	✓
<input type="checkbox"/>	10	test_crear_eventos_varios_9	10	✓	✓
<input type="checkbox"/>	11	test_crear_eventos_varios_10	10	✓	✓
<input type="checkbox"/>	12	test_actualizar_evento_valido	10	✓	✓
<input type="checkbox"/>	13	test_actualizar_evento_nombre_vacio	10	✓	✓
<input type="checkbox"/>	14	test_actualizar_evento_unicode	10	✓	✓
<input type="checkbox"/>	15	test_eliminar_evento_propietario	10	✓	✓
<input type="checkbox"/>	16	test_eliminar_evento_ojano	10	✓	✓
<input type="checkbox"/>	17	test_eliminar_evento_sin_seleccion	10	✓	✓
<input type="checkbox"/>	18	test_busqueda_nombre	10	✓	✓
<input type="checkbox"/>	19	test_busqueda_categoria	10	✓	✓
<input type="checkbox"/>	20	test_busqueda_precio_min_max	10	✓	✓
<input type="checkbox"/>	21	test_busqueda_fecha	10	✓	✓

6.3. Organización y Flujo de Trabajo

Branch protection rules

Add rule

main

Currently applies to 1 branch

Edit Delete

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feature/registro-y-filtrado	23 minutes ago		2	2	
develop	3 hours ago		2	2	
Pruebas	3 hours ago		2	2	
feature/implementar-sentry	yesterday		2	1	
feature/crud-eventos	2 days ago		2	0	

6.4. Integración de la plataformas con slack

Pruebas de Soft...

21 días restantes de prueba

Tu guía de Slack

Juntas

Canales

critical

general-pruebas-de-software

nuevo-canal

redes sociales

Añadir canales

Mensajes directos

carlos.arevalog

Daniel Zeballos tú

Invitar a otros

critical

Mensajes Sin título

Project: python Alert: python Short ID: PYTHON-11

Ver respuestas más recientes

Sentry APP 14:10

ha respondido a un hilo: Creando nuevo evento.

Creando nuevo evento.

Events: 136 State: Escalating First Seen: 14 hours, 22 minutes ago

Resolve Archive Select Assignee...

Project: python Alert: python Short ID: PYTHON-11

Enviar un mensaje a #critical

Pruebas de Soft...

21 días restantes de prueba

crítico

general-pruebas-de-software

nuevo-canal

redes sociales

+ Añadir canales

Mensajes directos

Carlos.arevalog

Daniel Zeballos

tú

Invitar a otros

Aplicaciones

GitHub

Mensajes

Acerca de

GitHub

APLICACIÓN

Este es el principio de tu historial de mensajes con @GitHub.

¿Cómo funciona GitHub?

Sábado, 6 de septiembre