****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *Juegos Florales - Pruebas Móviles***

Curso: TÓPICOS DE BASE DE DATOS AVANZADOS I

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

***Ronal Daniel Lupaca Mamani 2020067146***

***Jhonny Rivera Mendoza 2020067144***

**Tacna – Perú**

***2024***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema *{Nombre del Sistema}*

Documento de Especificación de Requerimientos de Software

Versión *{1.0}*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

**INDICE GENERAL**

[INTRODUCCION](#_Toc394513795) 4

[I. Generalidades de la Empresa](#_Toc394513799) 5

[1. Nombre de la Empresa 5](#_Toc394513800)

[2. Vision 5](#_Toc394513800)

[3. Mision 5](#_Toc394513800)

[4. Organigrama 5](#_Toc394513800)

[II. Visionamiento de la Empresa](#_Toc394513799) 5

[1. Descripcion del Problema 5](#_Toc394513800)

[2. Objetivos de Negocios 5](#_Toc394513800)

[3. Objetivos de Diseño 5](#_Toc394513800)

[4. Alcance del proyecto 5](#_Toc394513800)

[5. Viabilidad del Sistema 5](#_Toc394513800)

[6. Informacion obtenida del Levantamiento de Informacion](#_Toc394513800) 6

[III. Análisis de Procesos](#_Toc394513799) 6

[a) Diagrama del Proceso Actual – Diagrama de actividades](#_Toc394513800) 6

[b) Diagrama del Proceso Propuesto – Diagrama de actividades Inicial](#_Toc394513800) 7

[IV Especificacion de Requerimientos de Software](#_Toc394513799) 7

[a) Cuadro de Requerimientos funcionales Inicial](#_Toc394513800) 7

[b) Cuadro de Requerimientos No funcionales](#_Toc394513800) 7

[c) Cuadro de Requerimientos funcionales Final](#_Toc394513800) 8

[d) Reglas de Negocio](#_Toc394513800) 9

[V Fase de Desarrollo](#_Toc394513799) 12

[1. Perfiles de Usuario](#_Toc394513800) 12

[2. Modelo Conceptual 5](#_Toc394513800)

[a) Diagrama de Paquetes 5](#_Toc394513800)

[b) Diagrama de Casos de Uso](#_Toc394513800) 12

[c) Escenarios de Caso de Uso (narrativa)](#_Toc394513800) 14

[3. Modelo Logico](#_Toc394513799) 23

[a) Analisis de Objetos](#_Toc394513800) 23

[b) Diagrama de Actividades con objetos](#_Toc394513800) 32

[c) Diagrama de Secuencia](#_Toc394513800) 37

[d) Diagrama de Clases](#_Toc394513800) 42

[CONCLUSIONES](#_Toc394513803) 46

[RECOMENDACIONES](#_Toc394513804) 46

[BIBLIOGRAFIA](#_Toc394513805) 46

[WEBGRAFIA](#_Toc394513806) 46

# [INTRODUCCIÓN](#_heading=h.1hmsyys)

El presente documento tiene como objetivo detallar las pruebas móviles realizadas para la aplicación del proyecto "Juegos Florales". Este proyecto busca garantizar una experiencia óptima y eficiente para los usuarios de la Universidad Privada de Tacna al interactuar con la aplicación en eventos deportivos, culturales y académicos. Las pruebas se enfocan en verificar la funcionalidad, compatibilidad y desempeño de la aplicación móvil, asegurando su correcto funcionamiento en dispositivos Android de diferentes marcas y versiones.

El equipo de pruebas ha implementado herramientas modernas como Appium, Selenium y BrowserStack para automatizar y optimizar los procesos de prueba. A través de esta metodología, se garantiza un producto confiable y alineado con los estándares de calidad exigidos por la universidad.

# [Generalidades de la Empresa](#_heading=h.41mghml)

La Universidad Privada de Tacna (UPT) es una institución educativa reconocida por su compromiso con la formación integral de sus estudiantes y la promoción de actividades culturales, deportivas y académicas. En este contexto, la UPT organiza los "Juegos Florales", un evento que fomenta la participación activa de la comunidad universitaria en diversas disciplinas.

El proyecto de aplicación móvil se alinea con los objetivos de la universidad, brindando una herramienta tecnológica moderna que facilita el acceso a la información de los eventos y mejora la interacción entre los usuarios. Esta aplicación está diseñada para ser intuitiva, accesible y funcional, respondiendo a las necesidades específicas de la comunidad universitaria.

## [Nombre de la Empresa](#_heading=h.2grqrue)

* Universidad Privada de Tacna (UPT)SSALUD Calana - Tacna

## [Visión](#_heading=h.2grqrue)

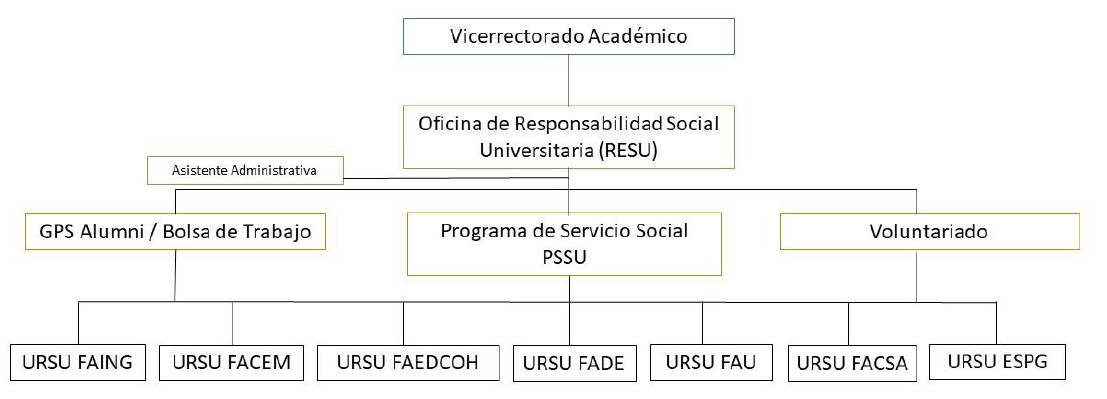
* Convertirse en una universidad líder a nivel regional y nacional en la implementación de tecnologías innovadoras que optimicen la gestión académica, cultural y deportiva, destacándose por su contribución al desarrollo integral de sus estudiantes.

## [Misión](#_heading=h.2grqrue)

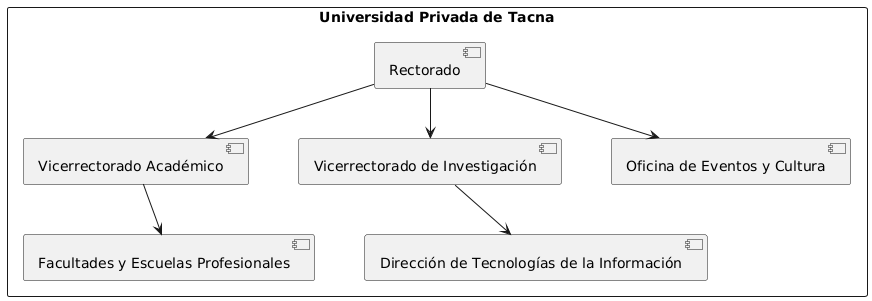
* Formar profesionales altamente capacitados y comprometidos con el desarrollo sostenible de la sociedad, ofreciendo una educación de calidad basada en valores, investigación e innovación tecnológica.

## [Organigrama](#_heading=h.2grqrue)

Este organigrama de la Universidad Privada de Tacna está compuesto por las siguientes áreas clave que apoyan el desarrollo de proyectos como "Juegos Florales":



Para los eventos florales :



# [Visiona miento de la Empresa](#_heading=h.41mghml)

## [Descripción del Problema](#_heading=h.2grqrue)

La Universidad Privada de Tacna enfrenta el desafío de gestionar de manera eficiente la organización y participación en sus eventos deportivos, culturales y académicos, como los "Juegos Florales". Actualmente, la comunicación y el seguimiento de actividades se realizan de forma manual o utilizando herramientas no centralizadas, lo que genera retrasos, falta de coordinación y poca claridad para los estudiantes, docentes y personal administrativo.

El problema radica en la ausencia de un sistema unificado que permita una gestión eficiente de los eventos, asegurando que la información esté actualizada, accesible y disponible en tiempo real para todos los involucrados.

## [Objetivos de Negocios](#_heading=h.2grqrue)

* **Garantizar la calidad del aplicativo móvil** mediante pruebas móviles automatizadas que validen la funcionalidad, compatibilidad y rendimiento.
* **Reducir costos y tiempos de pruebas manuales**, optimizando el ciclo de desarrollo y liberación de la aplicación.
* **Aumentar la satisfacción de los usuarios finales** al ofrecer una experiencia confiable y eficiente en todos los dispositivos Android soportados.
* **Proteger la imagen institucional** mediante la entrega de un aplicativo que cumpla con altos estándares de calidad tecnológica.

## [Objetivos de Diseño](#_heading=h.2grqrue)

* Diseñar un plan de pruebas móviles que abarque:
  + Validación de **interfaces de usuario** en distintos dispositivos.
  + Pruebas de **funcionalidades críticas** como el registro, notificaciones y visualización de eventos.
* Implementar un pipeline CI/CD que facilite pruebas continuas con herramientas como Appium, Selenium y BrowserStack.
* Asegurar la **compatibilidad multiplataforma**, abarcando versiones de Android desde 11.0 en adelante.
* Generar reportes detallados y visuales de los resultados de las pruebas mediante Allure.

## [Alcance del proyecto](#_heading=h.2grqrue)

El alcance incluye:

* **Automatización de pruebas móviles** para validar interfaces y funcionalidades críticas del aplicativo.
* Cobertura en pruebas tanto locales como en la nube (a través de BrowserStack) para garantizar la funcionalidad en una amplia variedad de dispositivos y versiones de Android.
* Validación del aplicativo en escenarios reales, como la inscripción a eventos y recepción de notificaciones.
* Generación de reportes de calidad y métricas de rendimiento para cada iteración.
* Capacitación básica al equipo de desarrollo para que pueda mantener y escalar las pruebas automatizadas.

## [Viabilidad del Sistema](#_heading=h.2grqrue)

En esta sección se analiza la viabilidad del sistema propuesto, evaluando su factibilidad técnica, económica y operativa.

* + - **Infraestructura tecnológica adecuada**: La universidad cuenta con recursos como servidores y conectividad para soportar las pruebas.
    - **Herramientas disponibles**: Uso de tecnologías reconocidas como Appium y BrowserStack, que permiten pruebas escalables y efectivas.
    - **Presupuesto adecuado**: Las herramientas y procesos utilizados son accesibles económicamente, ajustándose a las necesidades de un proyecto académico.
    - **Capacidad técnica del equipo**: El equipo de pruebas móviles tiene experiencia en el uso de frameworks de automatización, lo que asegura el éxito del proyecto.

## [Información obtenida del Levantamiento de Información](#_heading=h.2grqrue)

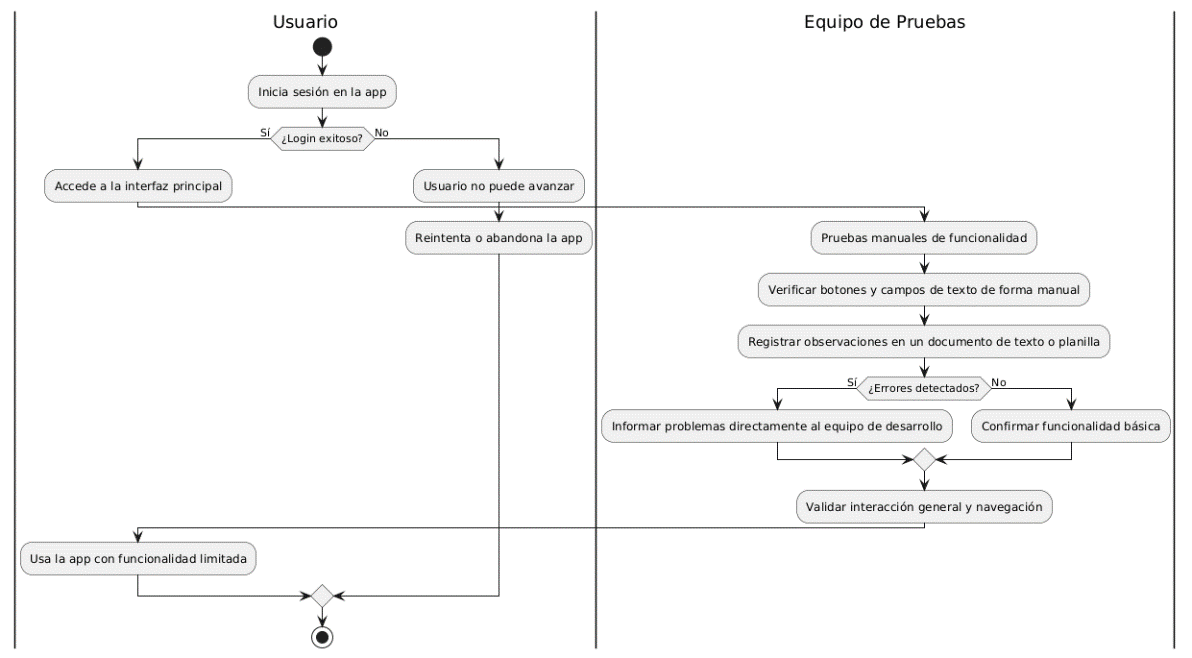
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo | Técnica | Sujeto | Tiempo | Información |
| Identificar funcionalidades críticas para probar | Entrevistas | Desarrolladores y usuarios finales | 1 semana | Registro de eventos, notificaciones, y rendimiento en dispositivos Android. |
| Identificar posibles fallos en el aplicativo | Pruebas exploratorias | Dispositivos Android variados | 2 semanas | Problemas de rendimiento en versiones antiguas de Android y fallas visuales. |
| Validar herramientas de pruebas automatizadas | Talleres técnicos | Equipo de TI | 1 semana | Herramientas aprobadas: Appium, Selenium, BrowserStack y Allure. |
| Establecer prioridades en las pruebas | Encuestas | Estudiantes y administrativos | 1 semana | Usuarios priorizan rapidez en el acceso a eventos y estabilidad de la aplicación. |

# [Análisis de Procesos](#_heading=h.41mghml)

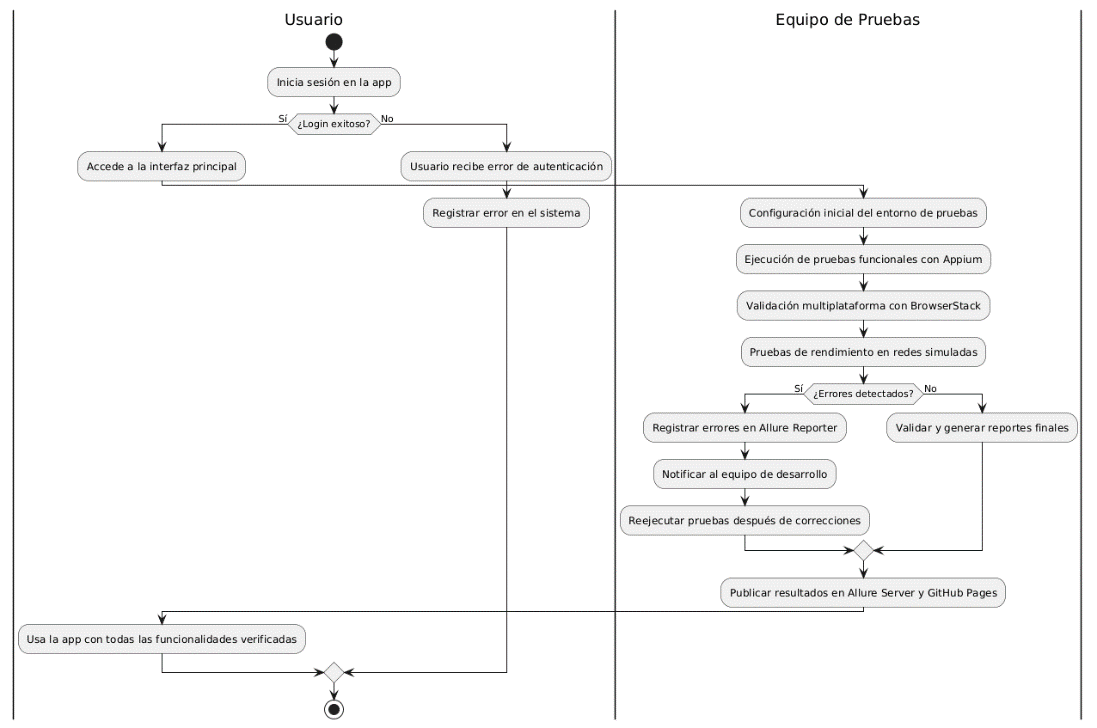
En esta sección se presentan los diagramas del proceso actual y propuesto según las herramientas que se usaran.

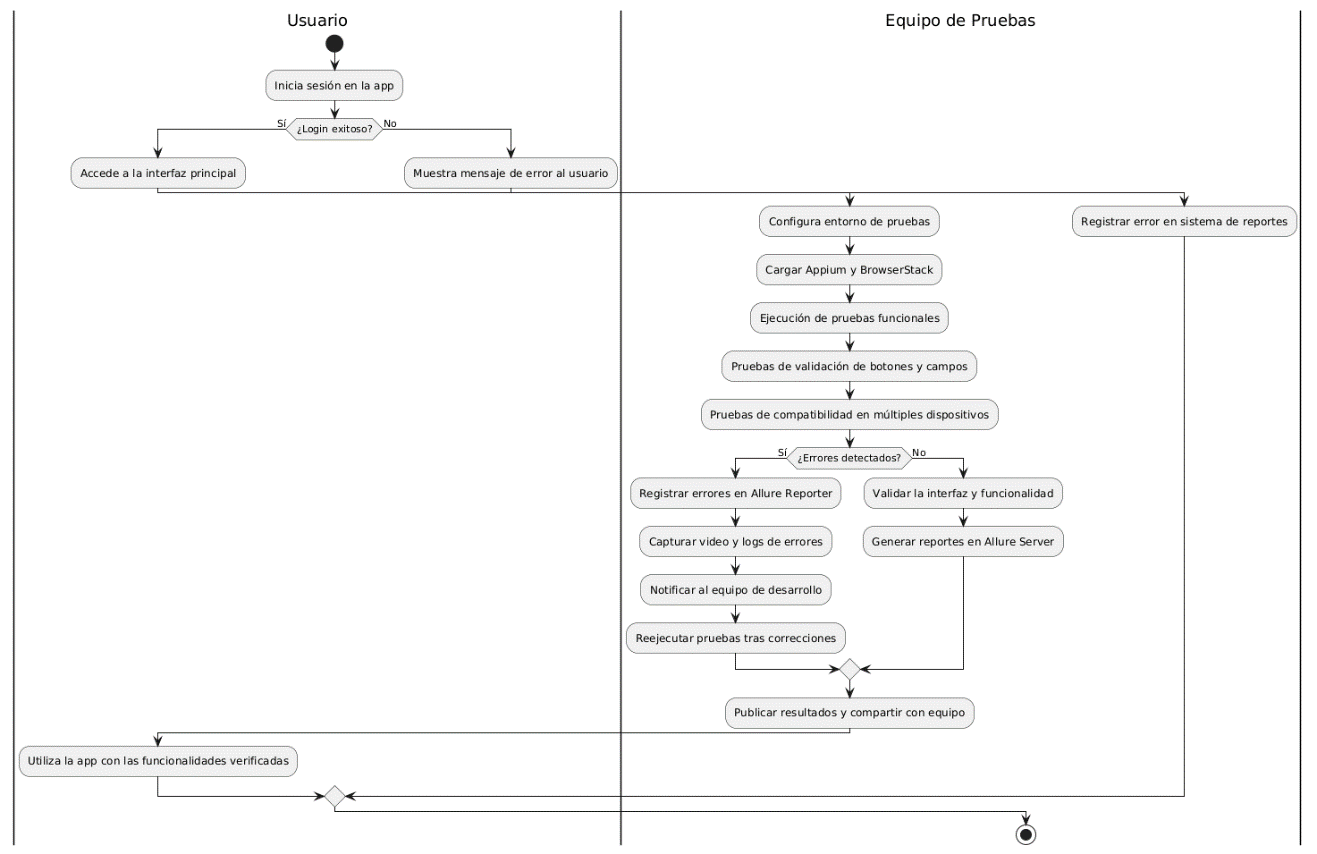
## [Diagrama del Proceso Actual – Diagrama de actividades](#_heading=h.2grqrue)

El proceso actual para garantizar la calidad del aplicativo móvil "Juegos Florales" es mayormente manual, lo que genera ineficiencias y problemas de cobertura. A continuación, se describe el flujo de actividades en el proceso actual:



## [Diagrama del Proceso Propuesto – Diagrama de actividades Inicial](#_heading=h.2grqrue)





# [Especificación de Requerimientos de Software](#_heading=h.41mghml)

Aquí se detallan los requerimientos funcionales y no funcionales del sistema de pruebas móviles propuesto, así como las reglas de negocio que guiarán su desarrollo. Estos requerimientos son fundamentales para definir qué funcionalidades debe tener el sistema de pruebas y cómo debe comportarse.

## [Cuadro de Requerimientos funcionales Inicial](#_heading=h.2grqrue)

Se presenta el cuadro inicial de requerimientos funcionales del software de pruebas móviles. Estos establecen las funcionalidades básicas que debe cumplir el sistema para realizar pruebas automatizadas, garantizar la compatibilidad y generar reportes detallados.

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Requerimiento | Descripción | Prioridad |
| RF-01 | Validación de inicio de sesión | Verificar que los usuarios puedan iniciar sesión en el sistema móvil con credenciales válidas. | Alta |
| RF-02 | Validación de botones | Comprobar que los botones de la interfaz respondan correctamente al ser pulsados. | Media |
| RF-03 | Validación de campos de texto | Garantizar que los campos de texto procesen entradas válidas y rechacen las inválidas. | Alta |
| RF-04 | Reportes de ejecución | Generar reportes detallados de las pruebas realizadas, incluyendo errores encontrados. | Alta |
| RF-05 | Compatibilidad de dispositivos | Verificar que la aplicación móvil funcione correctamente en diferentes dispositivos Android. | Alta |

## [Cuadro de Requerimientos No funcionales](#_heading=h.2grqrue)

Se especifican los requerimientos no funcionales del software de pruebas móviles. Estos se centran en aspectos como la seguridad, el rendimiento y la disponibilidad del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Requerimiento | Descripción | Prioridad |
| RNF-01 | Seguridad | Garantizar que las pruebas no expongan datos sensibles de los usuarios. | Alta |
| RNF-02 | Rendimiento | Asegurar que las pruebas se ejecuten en menos de 2 minutos por ciclo. | Alta |
| RNF-03 | Disponibilidad | El sistema de pruebas debe estar disponible el 99% del tiempo durante el proyecto. | Alta |
| RNF-04 | Interoperabilidad | Permitir pruebas en múltiples dispositivos y versiones de Android sin conflictos. | Alta |

## [Cuadro de Requerimientos funcionales Final](#_heading=h.2grqrue)

Se presenta el cuadro final de requerimientos funcionales refinados tras la revisión de las pruebas y ajustes necesarios.

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Requerimiento | Descripción | Prioridad |
| RF-01 | Validación mejorada de inicio | Revisar credenciales de usuario y su comportamiento en diferentes redes (WiFi, datos móviles). | Alta |
| RF-02 | Validación avanzada de botones | Asegurar que los botones realicen la acción correcta incluso en dispositivos de baja gama. | Media |
| RF-03 | Validación avanzada de texto | Implementar pruebas en campos con texto predictivo y autocompletado. | Alta |
| RF-04 | Integración con Allure | Generar reportes visuales detallados y automatizados con métricas de desempeño. | Alta |
| RF-05 | Pruebas en paralelo | Ejecutar pruebas en múltiples dispositivos simultáneamente usando BrowserStack. | Alta |

## [Reglas de Negoci](#_heading=h.2grqrue)o

Se describen las reglas de negocio que rigen el funcionamiento del sistema de pruebas móviles, especificando las restricciones y objetivos clave.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Tipo de Regla | Nombre de la regla | Descripción | Autoridad | Proceso Actual |
| RB-01 | Regla de estructura | Reportes automatizados | Todas las pruebas deben generar un reporte detallado en Allure automáticamente. | Equipo de QA(móvil) | Los reportes se generan manualmente tras las pruebas. |
| RB-02 | Regla de seguridad | Gestión de accesos | Los datos de las pruebas deben estar protegidos y accesibles solo al equipo autorizado. | Equipo de QA(móvil) | No existe control de acceso definido. |
| RB-03 | Regla de compatibilidad | Dispositivos Android | Todas las pruebas deben incluir al menos tres versiones diferentes de Android. | Equipo de QA(móvil) | Las pruebas no contemplan versiones anteriores. |

# [Fase de Desarrollo](#_heading=h.41mghml)

En esta sección se presentan los perfiles de usuario que interactúan con el aplicativo móvil de los "Juegos Florales". Los usuarios clave son el Usuario Final y el Equipo de Pruebas, ambos con roles esenciales para garantizar el correcto funcionamiento del sistema.

## [Perfiles de Usuario](#_heading=h.2grqrue)

**Perfil de Usuario: Usuario Final**

|  |  |
| --- | --- |
| Representante | Usuario Final |
| Descripción | Persona que utiliza el aplicativo para consultar resultados, eventos deportivos y culturales. |
| Tipo | Usuario |
| Responsabilidad | Probar funcionalidades básicas, como inicio de sesión, navegación y consultas. |
| Criterios de éxito | Acceder rápidamente a resultados y eventos deportivos sin errores. |
| Implicación | Garantizar una experiencia fluida y accesible para todos los usuarios. |

**Perfil de Usuario: Equipo de Pruebas**

|  |  |
| --- | --- |
| Representante | Equipo de Pruebas |
| Descripción | Grupo encargado de realizar pruebas automatizadas y funcionales en el aplicativo móvil. |
| Tipo | Equipo Técnico |
| Responsabilidad | Diseñar y ejecutar pruebas, validar resultados y generar reportes. |
| Criterios de éxito | Garantizar la compatibilidad y funcionalidad en múltiples dispositivos y versiones de Android. |
| Implicación | Asegurar que el aplicativo funcione de manera óptima, sin errores, y cumpla con los estándares de calidad. |

[Modelo Conceptual](#_heading=h.2grqrue)

### [Diagrama de Paquetes](#_heading=h.2grqrue)

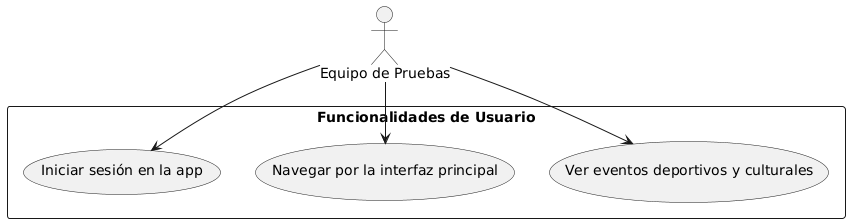
Interfaz de usuario gráfica, Aplicación

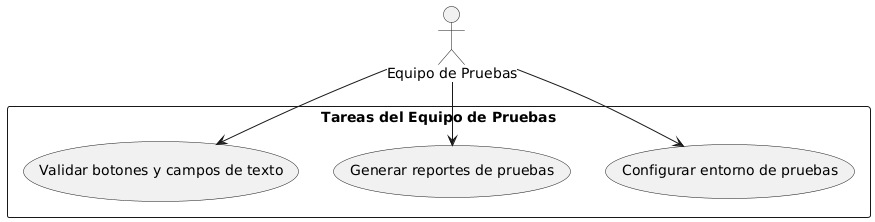
Descripción generada automáticamente

## [Diagrama de Casos de Uso](#_heading=h.2grqrue)

Casos de Uso Usuario y Equipo De Pruebas

Este diagrama ilustra los principales casos de uso para los usuarios del aplicativo móvil "Juegos Florales" de la Universidad Privada de Tacna. Los actores principales, las funcionalidades **Usuario** y el **Equipo de Pruebas**, interactúan con las funcionalidades diseñadas para garantizar una experiencia de usuario fluida y la calidad técnica del sistema.





* **Iniciar sesión en la app**: Es el punto de entrada para los usuarios finales, garantizando acceso seguro y personalizado a las funcionalidades del sistema.
* **Navegar por la interfaz principal**: Facilita una experiencia amigable, permitiendo a los usuarios explorar las funcionalidades principales de la app.
* **Ver eventos deportivos y culturales**: Esta funcionalidad destaca la relevancia del aplicativo para los "Juegos Florales", asegurando acceso a información actualizada.
* **Validar botones y campos de texto**: El equipo técnico verifica la correcta funcionalidad de los elementos interactivos, asegurando una experiencia sin errores.
* **Generar reportes de pruebas**: Documenta los resultados de las pruebas realizadas, ayudando al equipo de desarrollo a mejorar el sistema.
* **Configurar entorno de pruebas**: Asegura que las pruebas se realicen en un entorno controlado y adecuado, utilizando herramientas modernas como Appium y BrowserStack.Caso de Uso Personal Médico

## [Escenarios de Caso de Uso (narrativa)](#_heading=h.2grqrue)

**Iniciar sesión en la app. (Caso de Uso 01)**

|  |  |
| --- | --- |
| Nombre | Inicio de Sesión (Caso de Uso 01) |
| Actor | Usuario |
| Descripción | El equipo de pruebas verifica la funcionalidad del inicio de sesión en la aplicación móvil para los "Juegos Florales", asegurando que las credenciales del usuario sean validadas correctamente y que el sistema maneje tanto entradas válidas como erróneas de manera adecuada. |
| Flujo principal | |  |  | | --- | --- | | **Acción del Usuario** | **Respuesta del Sistema** | | 1. Accede a la página de inicio de sesión. | El sistema muestra una pantalla con el botón **"Iniciar sesión con Microsoft"** y un fondo simple. | | 2. Hace clic en el botón "Iniciar sesión con Microsoft". | El sistema redirige al usuario a la página de autenticación de Microsoft (OAuth), solicitando credenciales. | | 3. Ingresa su correo electrónico y contraseña de Microsoft. | Microsoft valida las credenciales ingresadas. Si son correctas, devuelve un token de acceso al sistema. Si son incorrectas, muestra un mensaje de error: "Correo o contraseña incorrectos. Inténtelo nuevamente." | | 4. Autoriza el acceso a la app (si es la primera vez). | Microsoft solicita autorización para permitir el acceso de la app a los datos del usuario (nombre y correo electrónico). Si el usuario acepta, el sistema procesa la solicitud. Si no, muestra un mensaje: "Acceso denegado por el usuario." | | 5. Finaliza el inicio de sesión. | Si las credenciales son correctas, el sistema autentica al usuario, crea la sesión y lo redirige al dashboard principal con un mensaje de bienvenida personalizado. | |
| Precondición | El usuario debe contar con una cuenta de Microsoft válida.  El sistema debe estar configurado para utilizar el protocolo OAuth con Microsoft. |
| Postcondición | El usuario accede correctamente al sistema y puede utilizar las funcionalidades.  Si las credenciales no son válidas o el usuario no autoriza el acceso, se le muestra un mensaje de error y no puede acceder al sistema. |

**Caso de Uso 02: Configurar entorno de pruebas**

|  |  |
| --- | --- |
| Nombre | Configurar entorno de pruebas (Caso de Uso 02) |
| Actor | Equipo de Pruebas |
| Descripción | El equipo de pruebas prepara el entorno para realizar las pruebas automatizadas y funcionales del aplicativo móvil, configurando herramientas como Appium y BrowserStack para ejecutar pruebas en dispositivos locales y en la nube. |
| Flujo principal | |  |  | | --- | --- | | Acción del Usuario | Respuesta del Sistema | | 1. Decide si configurará pruebas locales o en la nube. | 2.Se utiliza Appium para configuraciones locales y BrowserStack para pruebas en la nube. | | 3. Configura Appium para pruebas locales. | 4.El sistema verifica la instalación de Appium y la conexión con dispositivos físicos o emuladores locales. | | 5. Configura BrowserStack para pruebas en la nube. | 6. El sistema solicita las credenciales de BrowserStack (username y access key) y valida la autenticidad para acceder a los dispositivos virtuales en la nube. | | 7. Configura las librerías necesarias para el entorno. | 8.El sistema verifica la instalación de librerías como Selenium, pytest, allure-pytest y Appium Python Client, asegurando su correcto funcionamiento. | | 9. Verifica la conectividad entre el sistema y los dispositivos. | 10. Las herramientas comprueban que los dispositivos locales o en la nube están accesibles y listos para ejecutar las pruebas. | | 11. Genera un resumen de la configuración realizada. | 12. Las herramientas empleadas confirman los dispositivos configurados, las librerías instaladas y los entornos listos para pruebas, mostrando advertencias o errores que deben resolverse antes de ejecutar las pruebas. | |
| Precondición | El equipo de pruebas debe tener acceso a herramientas como Appium, BrowserStack, y los dispositivos necesarios.  Las librerías requeridas (e.g., Selenium, Appium Python Client) deben estar instaladas. |
| Postcondición | El entorno de pruebas está configurado y listo para ejecutar pruebas funcionales y automatizadas.  Cualquier error detectado en la configuración es registrado y notificado al equipo para su corrección. |

**Caso de Uso 03: Validar botones y campos de texto**

|  |  |
| --- | --- |
| Nombre | Validar botones y campos de texto (Caso de Uso 03) |
| Actor | Equipo de Pruebas |
| Descripción | El equipo de pruebas verifica la funcionalidad y la correcta interacción de los botones y campos de texto del aplicativo móvil en diferentes dispositivos y resoluciones, utilizando herramientas automatizadas y manuales. |
| Flujo principal | |  |  | | --- | --- | | Acción del Usuario | Respuesta del Sistema | | 1. Identifica los botones y campos de texto a validar. | El sistema carga las vistas correspondientes y resalta los elementos interactivos según lo especificado en el script de pruebas. | | 2. Ejecuta pruebas automatizadas sobre los botones y campos. | Las herramientas (e.g., Appium) simulan interacciones, como clics en botones y entradas de texto, verificando la respuesta del aplicativo a cada acción. | | 3. Realiza pruebas manuales complementarias en dispositivos específicos. | El sistema refleja en tiempo real las acciones realizadas manualmente, como validación de formatos de entrada, comportamiento al hacer clic en botones, y mensajes de error para entradas inválidas. | | 4. Valida la funcionalidad de botones para acciones específicas. | Se verifica que cada botón cumpla con su función esperada (e.g., navegación a otra pantalla, envío de información). | | 5. Comprueba los mensajes y validaciones en campos de texto. | El sistema muestra mensajes claros y adecuados para entradas incorrectas o vacías, y permite la entrada de datos correctos conforme a los requisitos. | | 6. Documenta los resultados de las validaciones. | Las herramientas generan reportes en tiempo real (e.g., Allure) con los resultados detallados, incluyendo capturas de pantalla y pasos ejecutados. | |
| Precondición | El usuario debe estar registrado y haber iniciado sesión en el sistema. |
| Postcondición | El usuario ha reservado una cita médica y tiene la opción de administrarla a través de su cuenta. |

**Generar reportes de pruebas (Caso de Uso 04)**

|  |  |
| --- | --- |
| Nombre | Generar reportes de pruebas |
| Actor | Paciente registrado |
| Descripción | El equipo de pruebas utiliza herramientas automatizadas para recopilar resultados de las pruebas realizadas en el aplicativo móvil y genera reportes detallados, incluyendo métricas, errores y capturas de pantalla. |
| Flujo principal | |  |  | | --- | --- | | Acción del Usuario | Respuesta del Sistema | | 1. Ejecuta las pruebas previamente configuradas. | 2.El sistema (e.g., Appium, BrowserStack) inicia las pruebas y recopila los datos de ejecución, como tiempos de respuesta, fallos, y comportamientos anómalos. | | 3. Recolecta datos de pruebas locales y en la nube. | 4.Las herramientas almacenan los resultados en archivos de registro y bases de datos temporales, diferenciando entre pruebas locales y remotas. | | 5. Configura la herramienta de generación de reportes (e.g., Allure). | 6.El sistema accede a los datos recopilados y prepara los reportes en un formato legible, incluyendo métricas clave, pasos ejecutados y capturas de pantalla relevantes. | | 7. Genera reportes automáticos al finalizar cada suite de pruebas. | 8.Allure genera un reporte detallado que incluye estadísticas globales de éxito y fallo, visualizaciones gráficas de resultados y una lista de errores detectados. | | 9. Revisa y valida los reportes generados. | 10.El equipo de pruebas verifica manualmente los reportes para asegurar que la información sea completa, precisa y relevante para los desarrolladores y otros interesados. | | 11. Comparte los reportes con el equipo de desarrollo. | 12. El sistema envía automáticamente los reportes por correo electrónico o los publica en un servidor accesible (e.g., Allure Server), asegurando que estén disponibles para todos los interesados. | |
| Precondición | Las herramientas de pruebas y reportes (e.g., Appium, Allure) deben estar configuradas y funcionando correctamente.  Los scripts de pruebas deben estar validados y listos para su ejecución. |
| Postcondición | Los reportes detallados son generados, revisados y compartidos con los equipos pertinentes.  Se identifican métricas clave y errores detectados para su posterior análisis y corrección. |

**Navegar por la interfaz principal (Caso de Uso 05)**

|  |  |
| --- | --- |
| Nombre | Interactuar con ChatBot (Caso de Uso 05) |
| Actor | Paciente |
| Descripción | El equipo de pruebas evalúa la navegabilidad de la interfaz principal del aplicativo móvil, asegurándose de que los elementos de la interfaz respondan correctamente y sean accesibles. |
| Flujo principal | |  |  | | --- | --- | | Acción del Usuario | Respuesta del Sistema | | 1. El equipo inicia el aplicativo móvil y accede a la interfaz principal. | 2. El sistema muestra la pantalla inicial con todas las opciones de navegación visibles. | | 3. El equipo interactúa con los menús y botones principales para verificar su funcionalidad. | 4. El sistema responde de manera fluida y redirige a las pantallas correspondientes sin errores. | | 5. El equipo prueba enlaces y accesos rápidos en la interfaz principal. | 6. El sistema registra las interacciones exitosas y genera reportes en caso de errores o tiempos de carga prolongados. | |
| Precondición | El aplicativo móvil debe estar instalado y ejecutándose correctamente en un dispositivo compatible. |
| Postcondición | La interfaz principal es validada como navegable y funcional según los requisitos establecidos. |

**Ver eventos deportivos y culturales (Caso de Uso 06)**

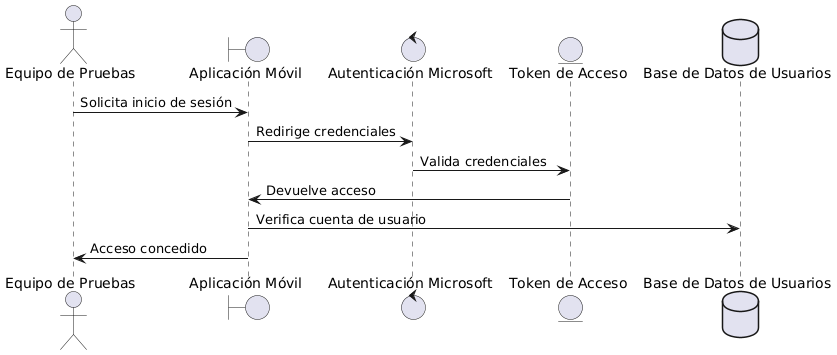
|  |  |
| --- | --- |
| Nombre | Gestión de la Cita Médica (Caso de Uso 06) |
| Actor | Equipo de Pruebas |
| Descripción | El equipo de pruebas verifica que el usuario pueda acceder y consultar información sobre eventos deportivos y culturales dentro del aplicativo móvil. |
| Flujo principal | |  |  | | --- | --- | | Acción del Usuario | Respuesta del Sistema | | 1. El equipo accede al apartado de eventos deportivos y culturales desde la interfaz principal. | 2. El sistema muestra una lista de eventos categorizados con sus detalles (fecha, hora, ubicación). | | 3. El equipo selecciona un evento para ver más detalles. | 4. El sistema despliega una pantalla con información específica del evento seleccionado, incluyendo opciones para agregar a favoritos o compartir. | | 5. El equipo prueba la funcionalidad de filtros (por categoría, fecha, ubicación). | 6. El sistema aplica los filtros y actualiza la lista de eventos en tiempo real. | |
| Precondición | La base de datos de eventos debe estar cargada en el sistema, y la interfaz debe estar correctamente configurada. |
| Postcondición | La funcionalidad para consultar eventos deportivos y culturales es validada y reportada como funcional o con observaciones para mejoras. |

# [3. Modelo Lógico](#_heading=h.41mghml)

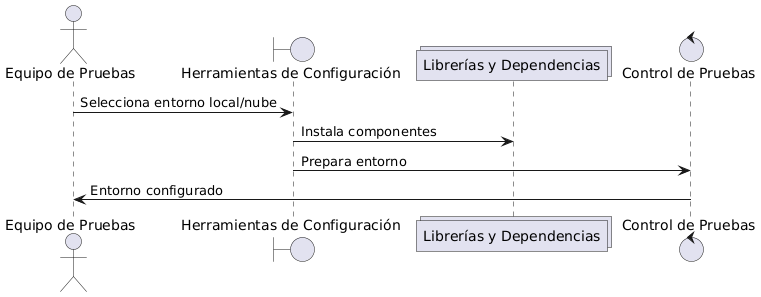
Este análisis tiene como objetivo proporcionar una comprensión detallada de los diversos procesos y flujos de trabajo involucrados en las pruebas.

## [Análisis de Objetos](#_heading=h.2grqrue)

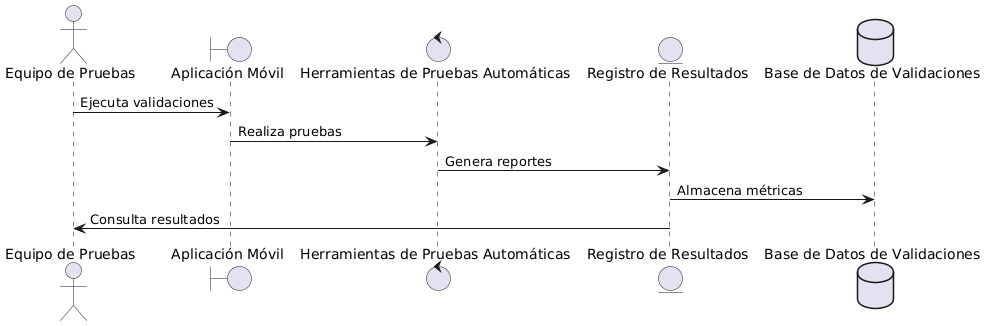
**Diagrama de Análisis Inicio de Sesión en la App**



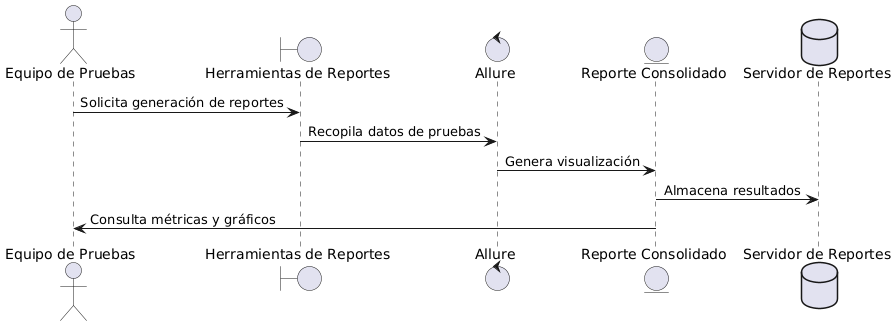
**Diagrama de Análisis Configurar Entorno de Pruebas**



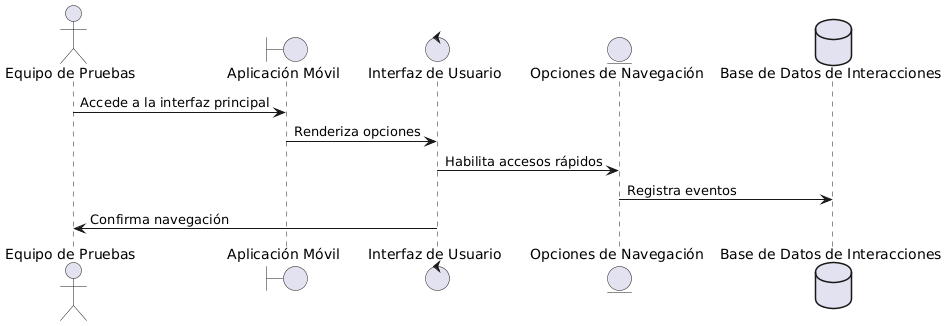
**Diagrama de Análisis Validar Botones y Campos de Texto**



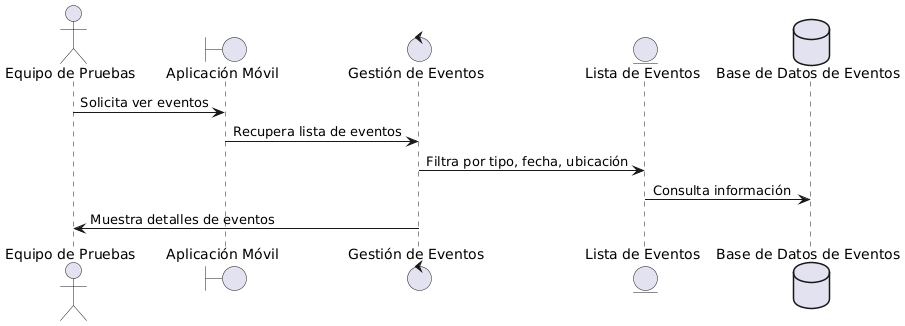
**Diagrama de Análisis Generar Reportes de Pruebas**



**Diagrama de Análisis Navegar por la Interfaz Principal**

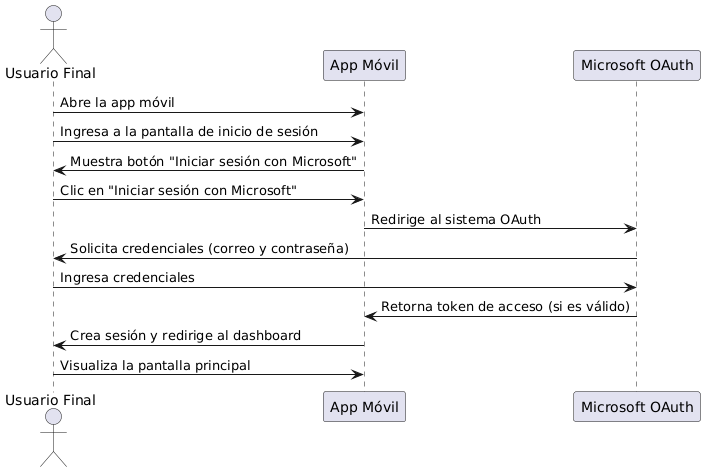
**

**Diagrama de Análisis Ver Eventos Deportivos y Culturales**

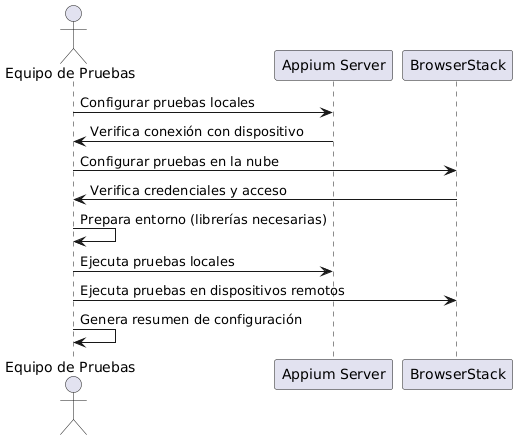


## [Diagrama de Secuencia](#_heading=h.2grqrue)

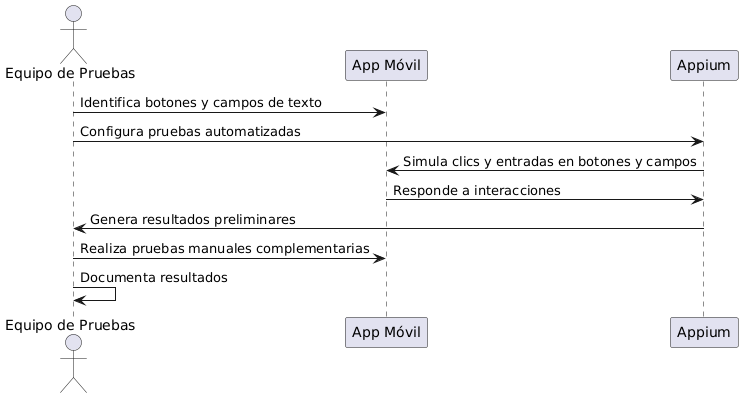
**Iniciar Sesión**



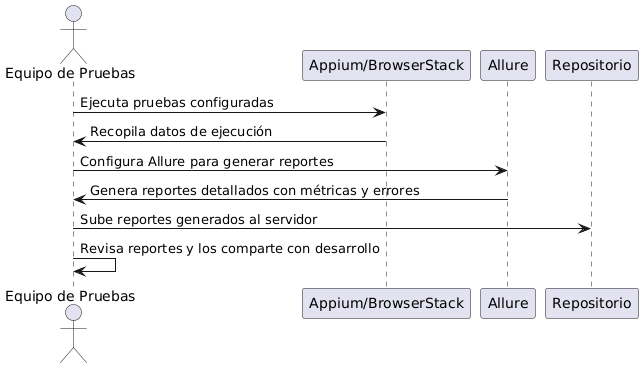
**Configurar Entorno de Pruebas**



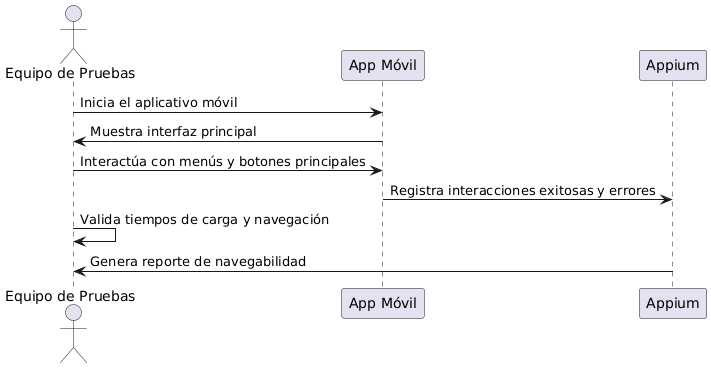
**Validar Botones y Campos de Texto**



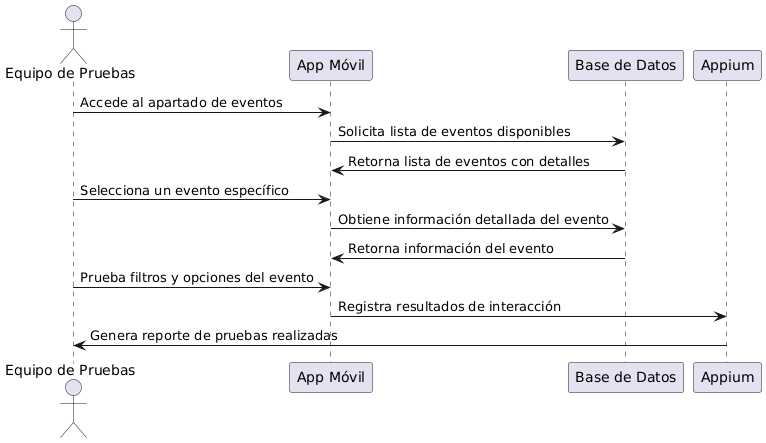
**Generar Reportes de Pruebas**



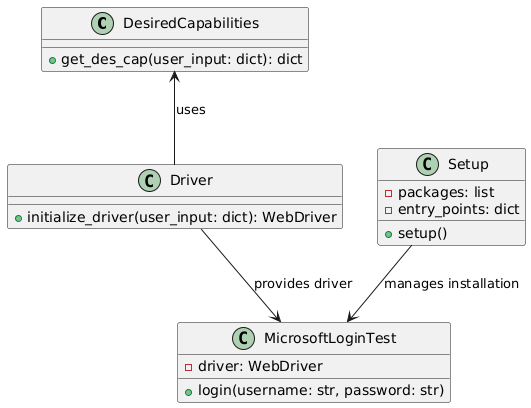
**Navegar por la Interfaz Principal**



**Ver Eventos Deportivos y Culturales**



## [Diagrama de Clases](#_heading=h.2grqrue)



# [CONCLUSIONES](#_heading=h.vx1227)

* La implementación de herramientas como Appium y BrowserStack permitió validar la funcionalidad, compatibilidad y rendimiento de la aplicación móvil "Juegos Florales" de manera eficiente, abarcando múltiples dispositivos y versiones de Android. Este enfoque automatizado optimizó el tiempo y los recursos invertidos en las pruebas, garantizando un producto final confiable y de alta calidad.
* Gracias a la integración de un pipeline CI/CD, se logró reducir significativamente los tiempos de ejecución de las pruebas, permitiendo la detección temprana de errores y facilitando la implementación de soluciones rápidas. Esto contribuyó a un flujo de trabajo más ágil y eficiente.
* Las pruebas realizadas aseguraron que el aplicativo móvil cumple con los estándares de calidad exigidos, brindando a los usuarios de la Universidad Privada de Tacna una experiencia fluida, confiable y accesible. Además, la colaboración entre el equipo de pruebas y el equipo de desarrollo se fortaleció gracias a la documentación detallada de los resultados, facilitando la identificación y corrección de errores.
* Finalmente, se destaca el impacto positivo de estas pruebas en la experiencia del usuario final, garantizando una navegación intuitiva y una funcionalidad estable en las principales características del aplicativo móvil.

# [RECOMENDACIONES](#_heading=h.3fwokq0)

* Es fundamental ampliar el alcance de las pruebas, incorporando dispositivos iOS y versiones futuras de Android para garantizar una compatibilidad multiplataforma y preparar el aplicativo para una base de usuarios más diversa. Asimismo, se recomienda mantener el pipeline CI/CD actualizado, revisando periódicamente las herramientas utilizadas y actualizando librerías y frameworks para evitar posibles incompatibilidades.
* La capacitación continua del equipo de pruebas es clave para mantener un alto nivel de eficiencia y calidad en el proceso. Introducir nuevas herramientas y técnicas de automatización permitirá al equipo enfrentar los desafíos tecnológicos de manera más efectiva.
* Además, se sugiere implementar herramientas de monitoreo post-producción que permitan detectar errores en tiempo real una vez que la aplicación esté en uso, asegurando una respuesta rápida y efectiva a cualquier problema que surja. Por último, se recomienda planificar iteraciones regulares para seguir optimizando el aplicativo conforme a las necesidades cambiantes de los usuarios, manteniendo su relevancia y funcionalidad a lo largo del tiempo.

# [BIBLIOGRAFÍA](#_heading=h.1v1yuxt)

* Appium Documentation. (2024). *Appium: Mobile App Automation Made Awesome*. Recuperado de: <https://appium.io>
* BrowserStack Documentation. (2024). *Cross Browser Testing Platform*. Recuperado de: <https://browserstack.com>
* Pytest Documentation. (2024). *Pytest Framework for Testing*. Recuperado de: <https://pytest.org>
* Allure Framework. (2024). *Allure Test Reporting Framework*. Recuperado de: https://docs.qameta.io/allure/
* Universidad Privada de Tacna. (2024). *Guía de desarrollo y calidad en proyectos tecnológicos*. Tacna, Perú.
* Bošković, D. (2022). *Automated Mobile Testing with Appium: A Comprehensive Guide*. Packt Publishing.