****

**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 Arquitectura 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

Contenido

[1. INTRODUCCIÓN 5](#_Toc69808834)

[1.1. Propósito (Diagrama 4+1) 5](#_Toc69808835)

[1.2. Alcance 5](#_Toc69808836)

[1.3. Definición, siglas y abreviaturas 5](#_Toc69808837)

[1.4. Organización del documento 5](#_Toc69808838)

[2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5](#_Toc69808839)

[2.1.1. Requerimientos Funcionales 5](#_Toc69808840)

[2.1.2. Requerimientos No Funcionales – Atributos de Calidad 5](#_Toc69808841)

[3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6](#_Toc69808842)

[3.1. Vista de Caso de uso 6](#_Toc69808843)

[3.1.1. Diagramas de Casos de uso 6](#_Toc69808844)

[3.2. Vista Lógica 6](#_Toc69808845)

[3.2.1. Diagrama de Subsistemas (paquetes) 7](#_Toc69808846)

[3.2.2. Diagrama de Secuencia (vista de diseño) 7](#_Toc69808847)

[3.2.3. Diagrama de Colaboración (vista de diseño) 7](#_Toc69808848)

[3.2.4. Diagrama de Objetos 7](#_Toc69808849)

[3.2.5. Diagrama de Clases 7](#_Toc69808850)

[3.2.6. Diagrama de Base de datos (relacional o no relacional) 7](#_Toc69808851)

[3.3. Vista de Implementación (vista de desarrollo) 7](#_Toc69808852)

[3.3.1. Diagrama de arquitectura software (paquetes) 7](#_Toc69808853)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 7](#_Toc69808854)

[3.4. Vista de procesos 7](#_Toc69808855)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 8](#_Toc69808856)

[3.5. Vista de Despliegue (vista física) 8](#_Toc69808857)

[3.5.1. Diagrama de despliegue 8](#_Toc69808858)

[4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 8](#_Toc69808859)

[Escenario de Funcionalidad 8](#_Toc69808860)

[Escenario de Usabilidad 8](#_Toc69808861)

[Escenario de confiabilidad 9](#_Toc69808862)

[Escenario de rendimiento 9](#_Toc69808863)

[Escenario de mantenibilidad 9](#_Toc69808864)

[Otros Escenarios 9](#_Toc69808865)

1. INTRODUCCIÓN
   1. Propósito (Diagrama 4+1)

El propósito del desarrollo pruebas móviles Juegos Florales es proporcionar una arquitectura robusta y eficiente para la automatización de pruebas en aplicaciones móviles, asegurando la calidad y funcionalidad del aplicativo. Este diseño se guía por el modelo 4+1, enfocándose en las siguientes vistas:

Vista lógica: Representación de los subsistemas y componentes esenciales.

Vista de implementación: Organización de módulos para pruebas automatizadas.

Vista de procesos: Flujo de actividades para validar funcionalidades críticas.

Vista física: Despliegue de herramientas y nodos de pruebas. Estas decisiones arquitectónicas están orientadas a cumplir los requerimientos funcionales, como la validación de inicio de sesión, y no funcionales, como la compatibilidad y el rendimiento.

* 1. Alcance

El alcance de este documento se centra en la definición de la arquitectura de software para el sistema de pruebas móviles Juegos Florales. Incluye:

* Validación de interfaces y funcionalidades críticas del aplicativo.
* Pruebas automatizadas y manuales en dispositivos Android.
* Generación de reportes detallados para el equipo de desarrollo. No incluye pruebas en sistemas operativos distintos a Android ni aspectos de diseño o desarrollo del aplicativo móvil.

* 1. Definición, siglas y abreviaturas
* Appium: Herramienta para pruebas automatizadas en aplicaciones móviles.
* BrowserStack: Plataforma para pruebas en dispositivos reales y emulados.
* CI/CD: Integración y Despliegue Continuo.
* Desired Capabilities: Configuración requerida para establecer una conexión con el dispositivo y la aplicación en Appium.
* APK: Android Package, archivo de instalación de aplicaciones para Android.
* UI Automator: Framework de pruebas automatizadas para Android.
  1. Organización del documento

El documento se organiza de la siguiente manera:

* Sección 1: Introducción al propósito, alcance y terminología del sistema.
* Sección 2: Detalle de los objetivos y restricciones arquitectónicas.
* Sección 3: Representación de la arquitectura mediante diagramas (casos de uso, clases, secuencia, paquetes y despliegue).
* Sección 4: Atributos de calidad del software, incluyendo funcionalidad, usabilidad, confiabilidad, rendimiento y mantenibilidad.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

* 1. Priorización de requerimientos

### Requerimientos Funcionales

|  |  |  |
| --- | --- | --- |
| ID | Descripcion | Prioridad |
| RF01 | Validación de inicio de sesión segura en la aplicación. | Alta |
| RF02 | Pruebas automatizadas para la interacción con botones. | Media |
| RF03 | Validación de los campos de texto con datos correctos. | Alta |
| RF04 | Generación de reportes detallados de ejecución. | Alta |

### 

### Requerimientos No Funcionales – Atributos de Calidad

*[Definir la prioridad de los requerimientos NO funcionales.]*

|  |  |  |
| --- | --- | --- |
| ID | Descripción | Prioridad |
| RNF01 | Garantizar la seguridad de los datos utilizados en las pruebas. | Alta |
| RNF02 | Asegurar que el tiempo de ejecución de pruebas no supere 2 minutos. | Alta |
| RNF03 | Disponibilidad continua del sistema (99% de tiempo activo). | Alta |
| RNF04 | Interoperabilidad con múltiples dispositivos y versiones de Android. | Alta |
| RNF05 | Generación automatizada de reportes visuales para facilitar el análisis. | Alta |

* 1. Restricciones

Restricciones Tecnológicas:

* + - * Uso exclusivo de herramientas de pruebas como Appium, Selenium y BrowserStack para garantizar la compatibilidad y escalabilidad.
      * Los dispositivos utilizados para las pruebas deben ser compatibles con Android 11 o superior.

Restricciones Económicas:

* + - * El presupuesto asignado debe cubrir las licencias de las herramientas utilizadas y el mantenimiento del entorno de pruebas.

Restricciones Operativas:

* + - * Las pruebas deben ejecutarse en un entorno controlado que cumpla con los estándares de seguridad establecidos.
      * Las pruebas automatizadas deben ser verificadas y aprobadas por el equipo de calidad antes de ser implementadas.

Restricciones Temporales:

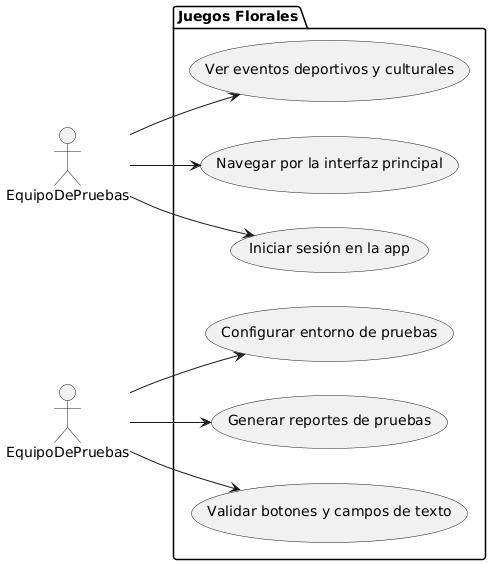
* + - * El sistema debe ser funcional y aprobado antes de la fecha de lanzamiento del aplicativo móvil "Juegos Florales".
      * La ejecución de pruebas debe ajustarse al cronograma definido, con un margen de error del 5%.

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso

La vista de caso de uso del sistema "Juegos Florales" tiene como objetivo describir las funcionalidades principales y los actores que interactúan con la aplicación. Estas funcionalidades representan las acciones más importantes que los usuarios y administradores pueden realizar dentro del sistema.

### Diagramas de Casos de uso



* 1. Vista Lógica

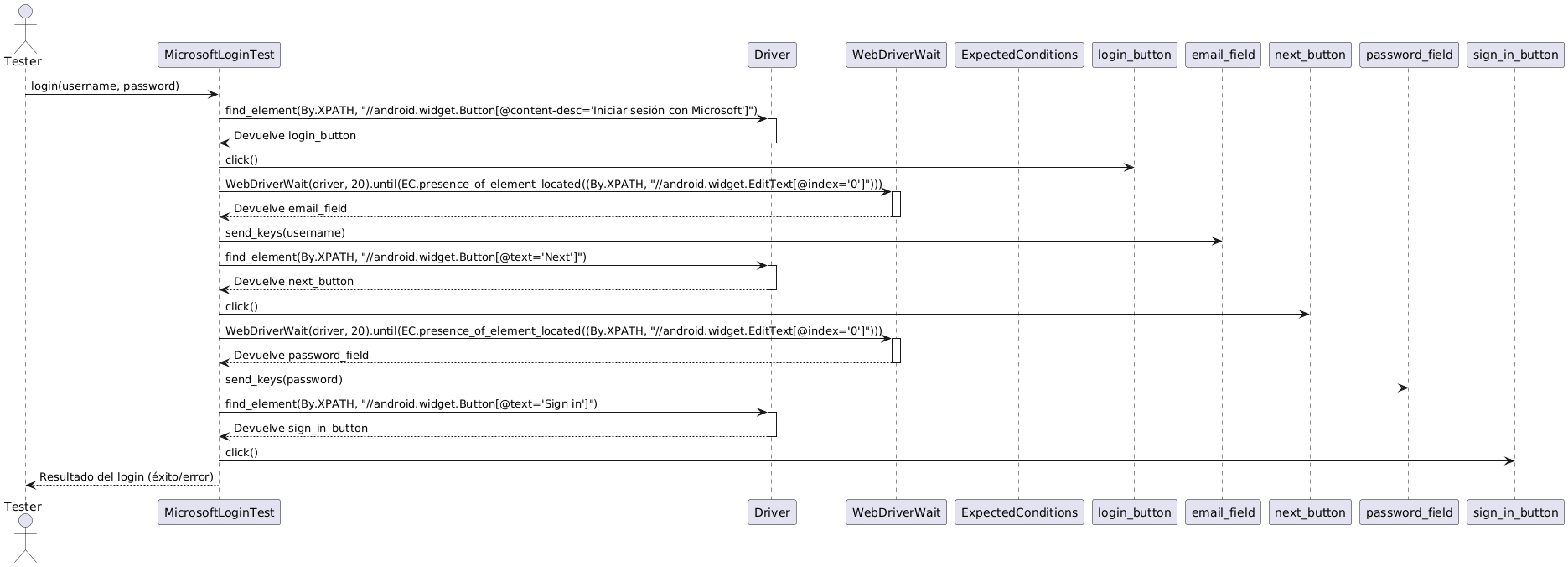
La vista lógica del sistema describe la estructura fundamental del diseño, incluyendo subsistemas y paquetes significativos. Esta vista se centra en satisfacer los requerimientos funcionales y facilita la interacción clara entre los componentes clave del sistema.

### Diagrama de Subsistemas (paquetes)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Diagrama de Secuencia (vista de diseño)

Diagrama de Secuencia (Pruebas al Aplicativo Móvil) "Validar el inicio de sesión mediante pruebas automatizadas"

### Diagrama de Clases

Diagrama

Descripción generada automáticamente

* 1. Vista de Implementación (vista de desarrollo)

### Diagrama de arquitectura

Diagrama

Descripción generada automáticamente

* 1. Vista de procesos

*[Describe la descomposición del sistema procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.]*

### Diagrama de Procesos del sistema (diagrama de actividad)

*[Se realizará un diagrama del o los procesos del sistema donde se exponga las actividades donde interviene el sistema propuesto, adicionando diagramas que definan el detalle la descomposición del sistema en procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican]*

* 1. Vista de Despliegue (vista física)

*[Se despliega uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos]*

### Diagrama de despliegue

*[un diagrama de despliegue, amplía el sistema de software y muestra los contenedores (aplicaciones, almacenamiento de datos, microservicios, etc.) que componen este sistema de software]*

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

Los Atributos de Calidad (QAs) son propiedades esenciales que permiten evaluar y medir el rendimiento, la funcionalidad y otros aspectos clave de un sistema, asegurando que este satisfaga las necesidades de los stakeholders. Aunque la funcionalidad del sistema es fundamental, los atributos de calidad no funcionales son críticos para garantizar la eficiencia, la seguridad y la confiabilidad del sistema en su contexto operativo..

Escenario de Funcionalidad

El sistema se evalúa en función de las características y capacidades que ofrece. En el caso de las pruebas al aplicativo móvil, la funcionalidad incluye:

Validación correcta de los datos de inicio de sesión.

Navegación fluida y sin errores dentro del aplicativo.

Seguridad general en la gestión de datos y credenciales.

Ejecución precisa y clara de cada prueba automatizada.

Escenario de Usabilidad

Este atributo evalúa la facilidad con la que los usuarios pueden aprender a utilizar el sistema y realizar sus tareas. Algunos aspectos relevantes son:

* **Aprendizaje rápido:** Los usuarios y testers deben comprender la interfaz y los flujos del sistema rápidamente.
* **Eficiencia de uso:** Reducción de errores durante las pruebas y generación de resultados claros.
* **Satisfacción del usuario:** Interacción sencilla y adaptada a las necesidades del equipo de pruebas.

Escenario de confiabilidad

La confiabilidad es fundamental para garantizar la robustez del sistema, abarcando:

* **Confidencialidad:** Protección de datos sensibles en pruebas de inicio de sesión y acceso.
* **Integridad:** Aseguramiento de que los datos procesados y almacenados no sean alterados durante las pruebas.
* **Disponibilidad:** El sistema debe estar accesible al menos el 99% del tiempo para ejecutar las pruebas sin interrupciones.

Escenario de rendimiento

El rendimiento se mide mediante la velocidad y eficiencia del sistema:

* **Velocidad de respuesta:** Tiempo mínimo para cargar interfaces y ejecutar pruebas.
* **Eficiencia en recursos:** Uso óptimo de memoria y CPU durante las pruebas en dispositivos Android.
* **Pruebas paralelas:** Capacidad para ejecutar múltiples pruebas simultáneamente usando herramientas como BrowserStack.

Escenario de mantenibilidad

La mantenibilidad garantiza que el sistema pueda adaptarse y escalarse fácilmente:

* **Extensibilidad:** Posibilidad de añadir nuevas pruebas automatizadas o funcionalidades sin afectar el sistema existente.
* **Adaptabilidad:** Ajustes rápidos a cambios en versiones de Android o actualizaciones de herramientas.
* **Diagnóstico y corrección:** Identificación rápida y resolución eficiente de errores.

Otros Escenarios

* **Performance:** Evaluación del tiempo de respuesta a eventos específicos y la capacidad de procesar múltiples eventos simultáneamente. La herramienta debe ofrecer reportes claros y detallados que ayuden a evaluar estos aspectos.
* **Escalabilidad:** El sistema debe soportar la adición de nuevos escenarios de prueba y dispositivos sin comprometer el rendimiento general.