****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto: Aplicativo Movil “MangaVerse”**

Curso: Soluciones Móviles I

Docente: Ing. Patrick Cuadros Quiroga

Integrantes:

**Villanueva Yucra, Josue (2018000722)**

**De La Cruz Choque, Rodrigo Martín (2019063328)**

**Cutipa Machaca, Arnold Felix (2019064022)**

**Perez Vizcarra, Juan José (2019063636)**

**Lira Alvarez, Rodrigo (2019063331)**

**Tacna – Perú**

***2023***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | RDC | RDC | PCQ | 12/12/2023 | Versión Original |
| 1.1 | JVY | RDC | PCQ | 12/12/2023 | Añadido de algunos Diagramas |

Proyecto: Aplicativo Movil “MangaVerse”

Documento de Arquitectura de Software

Versión 1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | RDC | RDC | PCQ | 12/12/2023 | Versión Original |
| 1.1 | JVY | RDC | PCQ | 12/12/2023 | Añadido de algunos Diagramas |

INDICE GENERAL

[1. Introducción 4](#_Toc153318650)

[1.1. Propósito 4](#_Toc153318651)

[1.2. Alcance 4](#_Toc153318652)

[1.3. Definición, siglas y abreviaturas 4](#_Toc153318653)

[1.4. Organización del documento 5](#_Toc153318654)

[2. Objetivos y Restricciones Arquitectónicas 5](#_Toc153318655)

[2.1. Requerimientos Funcionales 5](#_Toc153318656)

[2.2. Requerimientos No Funcionales 6](#_Toc153318657)

[3. Representación de la Arquitectura del Sistema 7](#_Toc153318658)

[3.1. Vista de Casos de Uso 7](#_Toc153318659)

[3.1.1. Diagrama de Casos de Uso 7](#_Toc153318660)

[3.2. Vista Lógica 8](#_Toc153318661)

[3.2.1. Diagrama de Subsistemas (paquetes) 8](#_Toc153318662)

[3.2.2. Diagrama de Secuencia (vista de diseño) 8](#_Toc153318663)

[3.2.3. Diagrama de Objetos 10](#_Toc153318664)

[3.2.4. Diagrama de Clases 11](#_Toc153318665)

[3.2.5. Diagrama de Base de datos (relacional o no relacional) 11](#_Toc153318666)

[3.3. Vista de Implementación (vista de desarrollo) 12](#_Toc153318667)

[3.3.1. Diagrama de arquitectura software (paquetes) 12](#_Toc153318668)

[3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes) 12](#_Toc153318669)

[3.4. Vista de procesos 13](#_Toc153318670)

[3.4.1. Diagrama de Procesos del sistema (diagrama de actividad) 13](#_Toc153318671)

[3.5. Vista de Despliegue (vista física) 14](#_Toc153318672)

[3.5.1. Diagrama de despliegue 14](#_Toc153318673)

[4. ATRIBUTOS DE CALIDAD DEL SOFTWARE 15](#_Toc153318674)

[Escenario de Funcionalidad 15](#_Toc153318675)

[Escenario de Usabilidad 15](#_Toc153318676)

[Escenario de confiabilidad 15](#_Toc153318677)

[Escenario de rendimiento 15](#_Toc153318678)

[Escenario de mantenibilidad 16](#_Toc153318679)

[Otros Escenarios 16](#_Toc153318680)

# Introducción

El mundo digital ha revolucionado la forma en que interactuamos con nuestras pasiones. En este contexto, MangaVerse emerge como una plataforma móvil diseñada para satisfacer la creciente demanda de los amantes del manga, ofreciendo una experiencia única y envolvente para los entusiastas de este arte japonés. Este Documento de Arquitectura de Software (SAD) representa la piedra angular que guiará el desarrollo de MangaVerse.

En este informe, exploraremos detalladamente los requisitos funcionales y no funcionales, así como las características clave que definirán la experiencia del usuario en MangaVerse. Además, delimitaremos el alcance del aplicativo, identificando a los usuarios objetivo y delineando el propósito fundamental de esta aplicación móvil.

A través de este documento, se busca establecer una hoja de ruta clara y precisa que permita el desarrollo de MangaVerse con la atención centrada en las necesidades y expectativas de una comunidad ávida por acceder al vasto universo del manga en la palma de sus manos.

## Propósito

El propósito de este documento es recoger, analizar y definir las necesidades de alto nivel y las características del manejo del catálogo de mangas proporcionados por el Proyecto, así como analizar su alcance.

## Alcance

Al momento de acceder a un manga para su lectura, la carga de datos y páginas son muy necesarias, una aplicación ágil permite una buena experiencia al usuario.

Una aplicación multiplataformas es algo muy requerido en cuanto a Disponibilidad; un punto fuerte en este tipo de aplicaciones al permitir una mayor variabilidad en su uso práctico.

## Definición, siglas y abreviaturas

* Aplicación Móvil: es un programa diseñado para un propósito específico que se descarga y se utiliza en ordenadores y dispositivos móviles como smartphones, tabletas, etc.
* Base de datos: Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático.
* Manga: término que se da a los cómics y novelas gráficas japonesas que contienen una secuencia narrativa.

## Organización del documento

En el proyecto tenemos diferentes roles que van a ocuparse de diferentes fases del desarrollo del proyecto. El equipo de trabajo está conformado por:

|  |  |
| --- | --- |
| NOMBRE | RESPONSABILIDADES |
| Villanueva Yucra, Josue | Desarrollador |
| De La Cruz Choque, Rodrigo Martín | Jefe de Proyecto |
| Cutipa Machaca, Arnold Felix | Analista |
| Pérez Vizcarra, Juan José | Administrador de configuración |
| Lira Alvarez, Rodrigo | Logistica |

El presente documento está conformado por las siguientes partes:

INTRODUCCIÓN: Contiene generalidades del sistema como el propósito, alcance del proyecto, abreviaturas y términos que se deben conocer para entender el proyecto.

OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS: Contiene todos los requerimientos, incluyendo su prioridad y también las restricciones.

REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA: Contiene todos los diagramas relacionados con la fase de diseño del sistema.

ATRIBUTOS DE CALIDAD DEL SOFTWARE: Contiene la información sobre los atributos de calidad que se tendrán en cuenta al implementar el sistema.

# Objetivos y Restricciones Arquitectónicas

## Requerimientos Funcionales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Nro* | *Requerimientos* | *Código* | *Descripción* | *Prioridad* |
| 1 | Registrar Usuario | RF01 | La aplicación debe permitir a los usuarios registrarse con información básica y acceder a su cuenta de MangaVerse. | Alta |
| 2 | Iniciar Sesión | RF02 | Los usuarios registrados podrán iniciar sesión en sus cuentas para acceder a contenido de MangaVerse. | Alta |
| 3 | Explorar Mangas | RF03 | Los usuarios deben poder explorar la biblioteca de MangaVerse, filtrar por el nombre del manga | Media |
| 4 | Perfil de Usuario | RF04 | La aplicación debe ofrecer a los usuarios la posibilidad de ver y editar su perfil, incluyendo información personal y preferencias de lectura. | Media |
| 5 | Visualizar novedades en página inicio | RF05 | Al iniciar sesión, los usuarios accederán a una página de inicio que muestre los animes populares basadas en sus preferencias de lectura y actividad previa. | Media |

## Requerimientos No Funcionales

|  |  |  |
| --- | --- | --- |
| ID | Requerimiento | Descripción |
| 1 | Usabilidad y Accesibilidad | La interfaz de usuario debe ser intuitiva, con navegación clara y fácil, asegurando la accesibilidad para usuarios con diferentes niveles de habilidad. |
| 2 | Escalabilidad | La aplicación debe ser capaz de manejar un aumento gradual en el número de usuarios concurrentes sin degradación significativa del rendimiento. |

# Representación de la Arquitectura del Sistema

Utilizamos MVVM para separar claramente la lógica de la interfaz de usuario (ViewModel) del modelo de datos y la vista. Esto tiene varios beneficios, como facilitar la prueba unitaria, mejorar la modularidad y permitir que diseñadores de UI y desarrolladores trabajen en paralelo sin afectar el funcionamiento de la aplicación.

Model (Modelo): El modelo representa los datos y la lógica de negocio de tu aplicación. En otras palabras, es la capa que maneja la obtención, manipulación y gestión de datos. Los modelos suelen ser objetos que encapsulan la información y la lógica necesaria para interactuar con los datos subyacentes, como bases de datos o servicios web.

View (Vista): La vista representa la interfaz de usuario (UI) de tu aplicación. Es la capa que muestra los datos al usuario y recibe las interacciones del usuario. En el contexto de las aplicaciones móviles o de escritorio, la vista puede ser una pantalla, ventana o página que el usuario ve y con la que interactúa.

ViewModel (Modelo de Vista): El ViewModel es una capa intermedia que actúa como intermediario entre el modelo y la vista. Su objetivo principal es proporcionar una representación de los datos del modelo en un formato que sea fácilmente consumible por la vista. Además, maneja las acciones del usuario y las transforma en acciones que afectan al modelo. El ViewModel también puede contener la lógica necesaria para la presentación de datos y la gestión de la interfaz de usuario.

## Vista de Casos de Uso

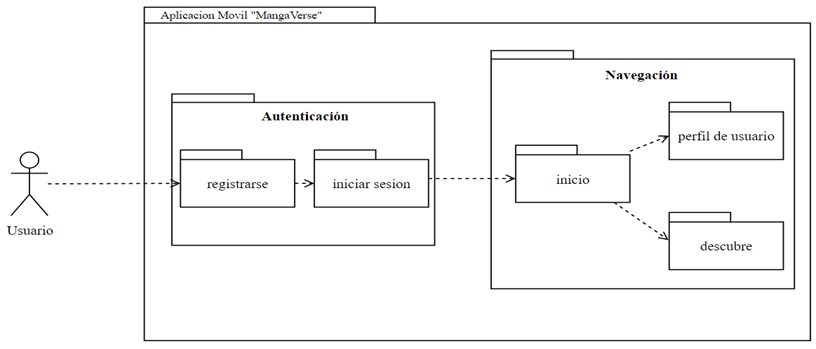
### Diagrama de Casos de Uso

Diagrama, Esquemático

Descripción generada automáticamente

## Vista Lógica

### Diagrama de Subsistemas (paquetes)



### Diagrama de Secuencia (vista de diseño)

* Iniciar sesion

Diagrama

Descripción generada automáticamente

* Registrar usuario

Diagrama

Descripción generada automáticamente

* Visualizar página de inicio

Diagrama, Tabla

Descripción generada automáticamente

* Visualizar página descubre

Tabla

Descripción generada automáticamente

* Visualizar página perfil

Diagrama

Descripción generada automáticamente

### Diagrama de Objetos

* Iniciar sesión

Diagrama

Descripción generada automáticamente

* Registrar usuario

Diagrama

Descripción generada automáticamente

### Diagrama de Clases

**Imagen que contiene interior, mapa, tabla, computadora

Descripción generada automáticamente**

### Diagrama de Base de datos (relacional o no relacional)

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

(Diagrama de base de datos no relacional - Firebase)

## Vista de Implementación (vista de desarrollo)

### Diagrama de arquitectura software (paquetes)

Diagrama

Descripción generada automáticamente

### Diagrama de arquitectura del sistema (Diagrama de componentes)

Diagrama

Descripción generada automáticamente

## Vista de procesos

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

Diagrama

Descripción generada automáticamente

## Vista de Despliegue (vista física)

### Diagrama de despliegue

Diagrama

Descripción generada automáticamente

# ATRIBUTOS DE CALIDAD DEL SOFTWARE

## Escenario de Funcionalidad

|  |  |
| --- | --- |
| Atributo de Calidad | Funcionalidad |
| Justificación | Un nuevo usuario es registrado a la Base de datos |
| Fuente | Interno al sistema |
| Estímulo | Registro de un nuevo usuario |
| Artefacto | Nodo del servidor Usuarios |
| Ambiente | Operación Normal |
| Respuesta | El sistema debe de registrar nuevo usuario |
| Medida de la respuesta | Cantidad de usuarios registrados |

## Escenario de Usabilidad

|  |  |
| --- | --- |
| Atributo de Calidad | Usabilidad |
| Justificación | Se hace uso de la interfaz de la aplicación |
| Fuente | Interna al Sistema |
| Estímulo | Interfaz de la Aplicación |
| Artefacto | Interfaz de usuario |
| Ambiente | Operación Normal |
| Respuesta | La aplicación dispone de una interfaz intuitiva que permite la rápida manipulación de esta |
| Medida de la respuesta | Celeridad del uso de la aplicación |

## Escenario de confiabilidad

|  |  |
| --- | --- |
| Atributo de Calidad | Confiabilidad |
| Justificación | El sistema se cae por un corte de energía |
| Fuente | Externo |
| Estímulo | Servicio no disponible |
| Artefacto | Servidor |
| Ambiente | Corte de energía |
| Respuesta | El sistema debe de funcionar sin errores al finalizar el corte de energía |
| Medida de la respuesta | Cantidad de errores por cortes de energía |

## Escenario de rendimiento

|  |  |
| --- | --- |
| Atributo de Calidad | Rendimiento |
| Justificación | La aplicación necesita responder apropiadamente al mostrar la lista buscada |
| Fuente | Interna al Sistema |
| Estímulo | Listar resultados |
| Artefacto | Dispositivo |
| Ambiente | Operación Normal |
| Respuesta | La aplicación despliega rápidamente los datos consultados |
| Medida de la respuesta | Velocidad de respuesta |

## Escenario de mantenibilidad

|  |  |
| --- | --- |
| Atributo de Calidad | Mantenibilidad |
| Justificación | Uso apoyado en una API externa |
| Fuente | Externo al sistema |
| Estímulo | Uso de API’s |
| Artefacto | Nodo del Sistema - Listar |
| Ambiente | Operación Normal |
| Respuesta | El sistema debe de tener siempre sus datos en regla |
| Medida de la respuesta | Cantidad de Errores prevenidos |

## Otros Escenarios

|  |  |
| --- | --- |
| Atributo de Calidad | Disponibilidad |
| Justificación | La aplicación debe de funciona en cualquier hora |
| Fuente | Usuario |
| Estímulo | Inicio de la Aplicación en cualquier hora del dia |
| Artefacto | Sistema |
| Ambiente | Operación Normal |
| Respuesta | La aplicación debe de funcionar en cualquier momento |
| Medida de la respuesta | Errores en los logs de error al iniciar aplicación |