Diseño e implementación de un prototipo basado en realidad virtual como herramienta de apoyo para la enseñanza de cálculo multivariado, para estudiantes de ingeniería de sistemas y computación de la Universidad del Quindío

Documento de arquitectura de software

Versión 1.0

Historial de documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 14-10-2016 | 1.0 | Definición inicial del documento | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P. |
| 15-10-2016 | 1.0 | Definición de casos de uso | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P. |
| 16-10-2016 | 1.0 | Definición de vista implementación | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P. |
| 16-10-2016 | 1.0 | Definición de vista despliegue | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P. |
| 16-10-2016 | 1.0 | Definición de vista lógica | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P. |

Contenido

1. Introducción 4

1.1 Propósito de este documento 4

1.2 Referencias 4

1.3 Resumen del documento 4

2. Requerimientos de arquitectura 6

2.1 Requerimientos no funcionales 6

2.2 Vistas de casos de uso (requerimientos funcionales) 6

3. Vista lógica 7

3.1 Niveles 7

3.2 Realización de casos de uso 7

4. Vista de implementación 8

Ilustración 2: Lógica de movimiento 9

Ilustración 3: Funciones Permitidas

Ilustración 4: Interacción con los objetos 10

4.1 Estructura de los paquetes 10

4.2 Realización de niveles 10

4.3 (Re)uso de componentes y marcos de trabajo 11

5. Vista de despliegue 11

# Introducción

## Propósito de este documento

El presente documento define y describe la implementación del Software del proceso de aseguramiento de la calidad del software (*Software Quality Assurance,* SQA), en el proyecto Implementación de un prototipo basado en realidad virtual como herramienta de apoyo para la enseñanza de cálculo multivariado y vectorial, para estudiantes de ingeniería de sistemas y computación de la Universidad del Quindío. El propósito aquí descrito es actualizado y revisado periódicamente por todos los miembros del equipo, para garantizar que los cambios requeridos sean incorporados en el mismo.

## Referencias

|  |  |  |  |
| --- | --- | --- | --- |
| **Titulo** | **Versión** | **Autor** | **Locación** |
| Visión | 1.2 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Documento de especificación de casos de uso | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Diagramas de casos de uso | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Patrones de diseño | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Estimación | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| SCM-PMC | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Plan SQA | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Procesos SQA | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Planeación Iteración | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Planeación Riesgos | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |
| Vistas 4 + 1 | 1.0 | Paulo C. Alvis R.; Leidy A. Raigoza P.; Diego A. Sánchez P | https://github.com/andrearaigoza/Documentos-Software |

## Resumen del documento

| **Capitulo** | **Lector** | **Objectivo** |
| --- | --- | --- |
| 2 Requerimientos arquitectonicos | Arquitecto de Software | Resumen de la relevancia de los requerimientos arquitectónicos. |
| 3 Vista lógica | Desarrollador | Conocimiento de la estructura conceptual de la aplicación, como la base para diseño técnico. |
| 4 Vista de implementación | Desarrollador | Conocimiento de la estructura técnica de la aplicación. |
| 5 Vista de despliegue | Administradores de sistema | Knowledge of the way in which the application is deployed and (internal and external) communication takes place.  Conocimiento de la manera como la aplicación es desplegada y (interna y externamente) como las comunicaciones tienen lugar. |

# Requerimientos de arquitectura

Describen específicamente las necesidades que debe suplir el aplicativo a través del análisis previo que debe realizar el equipo de desarrollo al inicio del proyecto en conjunto con los stakeholders en especial el usuario final quien será el que utilice el producto final

## Requerimientos no funcionales

Los requerimientos no funcionales son categorizados por seguridad, desempeño, escalabilidad, disponibilidad, integridad, entre otros. En resumen son aquellos que describen como el software debe comportarse, es decir cómo debe hacer algo, más no qué hacer.

| **Source** | **Name** | **Architectural relevance** | **Addres­sed in:** |
| --- | --- | --- | --- |
| Visión | Desempeño | El nivel de desempeño del aplicativo en múltiples dispositivos indicará un eficiente uso de los recursos hardware de los cuales depende éste para funcionar | 1 |
| Stakeholder | Usabilidad | Hacer posible un fácil uso del aplicativo proporcionando una buena experiencia de usuario es crucial para el proyecto ya que se trata de un total contenido gráfico. | 2 |
| Visión | Portabilidad | Permitir su fácil instalación y ejecución en diversos dispositivos contribuye a la masificación de su uso por parte del usuario final. | 3 |
| Visión | Calidad | La implementación de buenas prácticas y estándares de producción en el desarrollo del proyecto garantiza un producto con estándares más altos de calidad | 4 |
| Stakeholder | Interfaz | Describe la apariencia del producto, especificando como se presenta el producto al usuario final en su totalidad según contexto y análisis de mercado. | 5 |
|  |  |  |  |

## Vistas de casos de uso (requerimientos funcionales)

A continuación se listan los requerimientos funcionales con los que debe contar el aplicativo, según lo establecido con los Stakeholders.

| **Source** | **Name** | **Architectural relevance** | **Addressed in:** |
| --- | --- | --- | --- |
| Especificación caso de uso realizar rotacion | Realizar rotación | Desde el punto de vista de la arquitectura se asocia este requerimiento funcional a la parte de usabilidad e interfaz, de la forma como se le presentan los escenarios virtuales al usuario final. | 3.3 |
| Especificación caso de uso realizar traslacion | Realizar traslacion | Desde el punto de vista de la arquitectura se asocia este requerimiento funcional a la parte de usabilidad e interfaz, de la forma como se le presentan los escenarios virtuales al usuario final. | 3.3 |
| Especificación caso de uso realizar interacción | Realizar interacción | Desde el punto de vista de la arquitectura se asocia este requerimiento funcional a la parte de usabilidad e interfaz, de la forma como se le presentan los escenarios virtuales al usuario final. | 3.3 |
|  |  |  |  |

# Vista lógica

Esta seccion se enfocará en estructurar la vista logica acorde a la arquitectura planteada al inicio. De esta manera se justificará cada elemento logico debido a que soporta algun componente arquitectinico.

## Niveles

A continuación se relacionan las responsabilidades del sistema planteado a través del modelo 4-tier que de manera gráfica muestra la forma en que se relacionan los componentes.

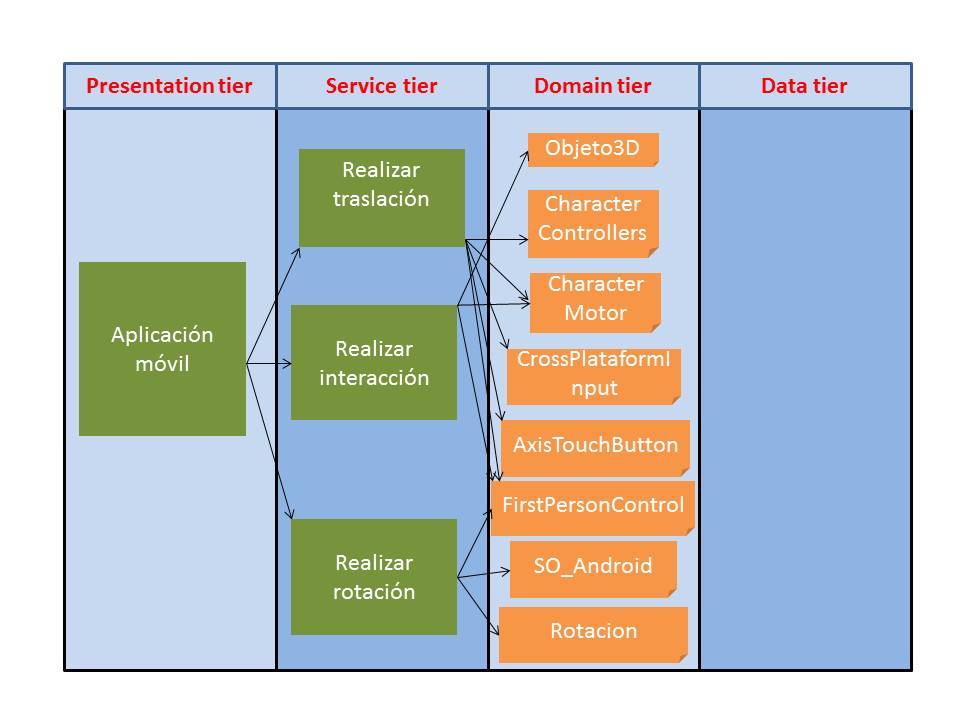


Figura 1: Modelo 4-Tier

## Realización de casos de uso

Anexo a este documento se encuentran los diagramas que exponen la realización de cada caso de uso, además se incluyen los documentos de especificación de casos de uso.

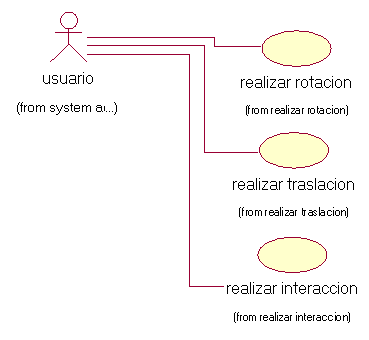


Figura 2: Casos de uso

# Vista de implementación

La implementacion técnica se centra en la funcionalidad que la aplicación ofrece al usuario final.

En la Ilustración 1 se muestra la estructura que soporta las funcionalidades de la aplicación, que incluyen el movimiento iniciado por el usuario y la interacción que puede tener con los elementos gráficos que el usuario puede visualizar.

En la Ilustración 2, Ilustración 3 se pueden visualizar las funcionalidades que son soportadas por la aplicación a través de una implementación lógica acorde al hardware, software y necesidades del proyecto, lo que incluye movimientos de rotación y traslación necesarios para que el usuario navegue por los diferentes escenarios de la aplicación, como también la posibilidad de interactuar con algunos de los objetos que se encuentran en dichos escenarios.

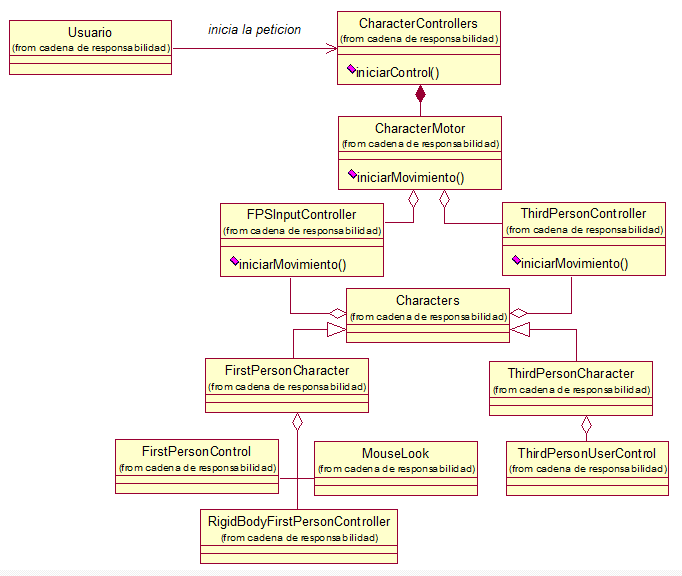


Ilustración 2: Lógica de movimiento

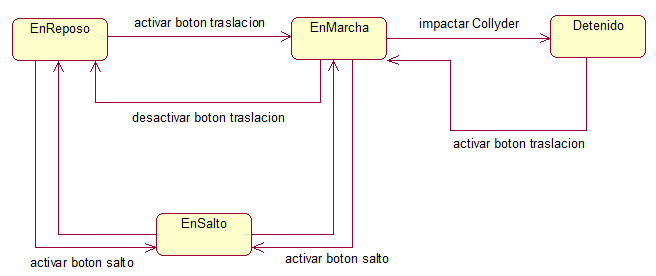


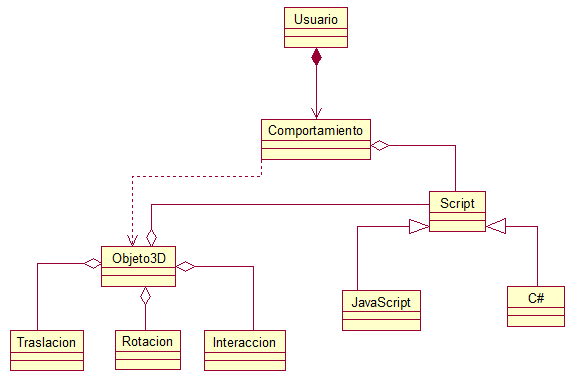
Ilustración 3: Funciones Permitidas 

Ilustración 4: Interacción con los objetos

## Estructura de los paquetes

A continuación se describe la estructura de los paquetes creados al momento de instalar la aplicación en un smartphone. En la siguiente ruta:

* Internal storage > Android > data > com.uniquindio.vrpad > cache > UnityShaderCache

## Realización de niveles

A continuación se describe la realización de los niveles identificados anteriormente, especificando las reglas que aplican a cada componente.

* Traslación: entiendase para este nivel la capacidad del usuario para moverse de manera bidimensinal, es decir, hacia adelante, atrás, izquierda y derecha. Para lograr esta funcionalidad es necesario contar con un periferico de entrada Bluetooth que permita dar dirección mediante palanca o botones.
* Rotación: para lograr esta funcionalidad es necesario contar con un dispositivo para visualización tridimensional, como lo son los lentes de realidad virtual. Es totalmente necesario contar con un smartphone que tenga incorporado un sensor giroscopio para lograr la rotación en eje X como en el eje Y.
* Interacción: algunos de los objetos presentados en las escenas de la aplicación cuentan con la opción de ser transformados en tiempo real por parte del usuario, pero éstas estarán limitadas a cambios como en posicion y tamaño.

## (Re)uso de componentes y marcos de trabajo

Una pequeña sección de componentes y frameworks que no han sido especificados en las secciones anteriores pero que son reutilizados durante todo el desarrollo de la aplicación son objetos 3D como planos, cilindros con sus respectivos materiales, las escenas base para la implementación de la realidad virtual la cual hace posible la interacción con el smartphone y los lentes. De la misma forma se cuenta con una base definida para la animación de algunos sólidos según lo requieran para la explicación de fenomenos matematicos.

# Vista de despliegue

Para el despliegue de la aplicación solo se contará con un nodo que en este caso será un Smartphone Xiaomi Mi 3w, cuya configuración de hardware y software se describirá a continuación

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| Pantalla | Nodo | LCD IPS touchscreen capacitivo, 16M colores. 1080 x 1920 pixeles, 5 pulgadas.   * Pantalla Gorilla Glass 3 * Soporte multitouch * Sensor acelerómetro para auto rotación * Sensor de proximidad para auto apagado * Barómetro * Sensor de luz ambiente * Sensor giroscopio * Video 1080p |
| Procesador | Nodo | Qualcomm MSM8274AB Snapdragon 800 quad-core 2.3GHz, GPU Adreno 330. |
| Sistema Operativo | Nodo | Android OS, V4.3 Jelly Bean – MIUI V5 |
| Bluetooth | Nodo | V4.0 A2DP, LE |
| Batería | Nodo | Standard, Li-Ion 3050 mAh |
| Memoria | Nodo | 2 GB RAM – 16 GB almacenamiento interno |