Grupo VLC.

FCEFyNApp.
Documento de Arquitectura de Software.

Autores: López Gastón, Vignolles Iván, Lamberti Germán.
Versión del documento: 1.0.0
Materia: Ingeniería del Software.
Profesor: Nonino, Julián.
Facultad de Ciencias Exactas, Físicas y Naturales.
Universidad Nacional de Córdoba.
Año 2017.

Historial de revisiones.

Versión	Fecha	Resumen de cambios.	Autor/es
1.0.0	28/05/2017	Documento inicial.	López Gastón. Vignolles Iván. Lamberti Germán.

Índice.

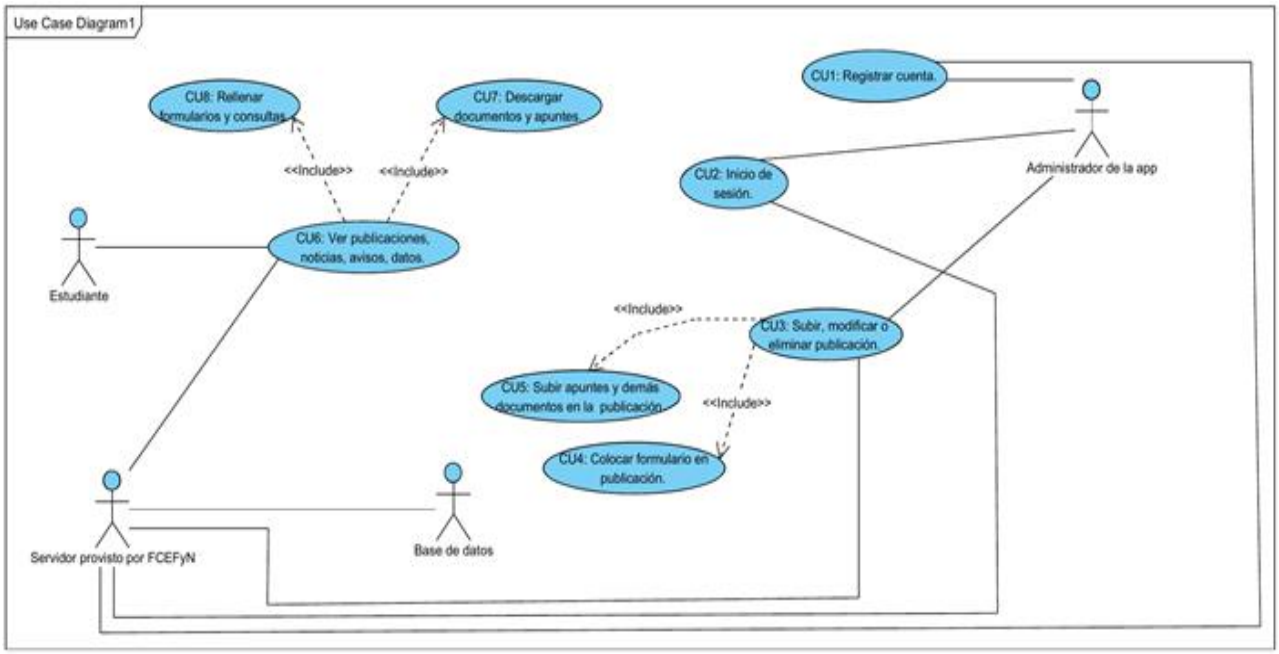
Documento de arquitectura de software.

1. Introducción.

1.1. Objetivos del documento.

Este documento es actualizable de acuerdo a los cambios técnicos de arquitectura que vayan apareciendo. El objetivo del documento es mantener organizada la arquitectura técnica organizacional que va a presentar el sistema a desarrollar y proveer una fuente de referencia para los analistas y diseñadores de la aplicación. Estos últimos utilizarán este documento para propósitos informativos de entendimiento de la actual arquitectura, y para los propósitos del análisis con el fin de discernir si la arquitectura, como es, apoyará los requisitos de la aplicación. Todos los cambios aplicables a la arquitectura deberán ser aprobados por las dos terceras partes del total del equipo VLC.

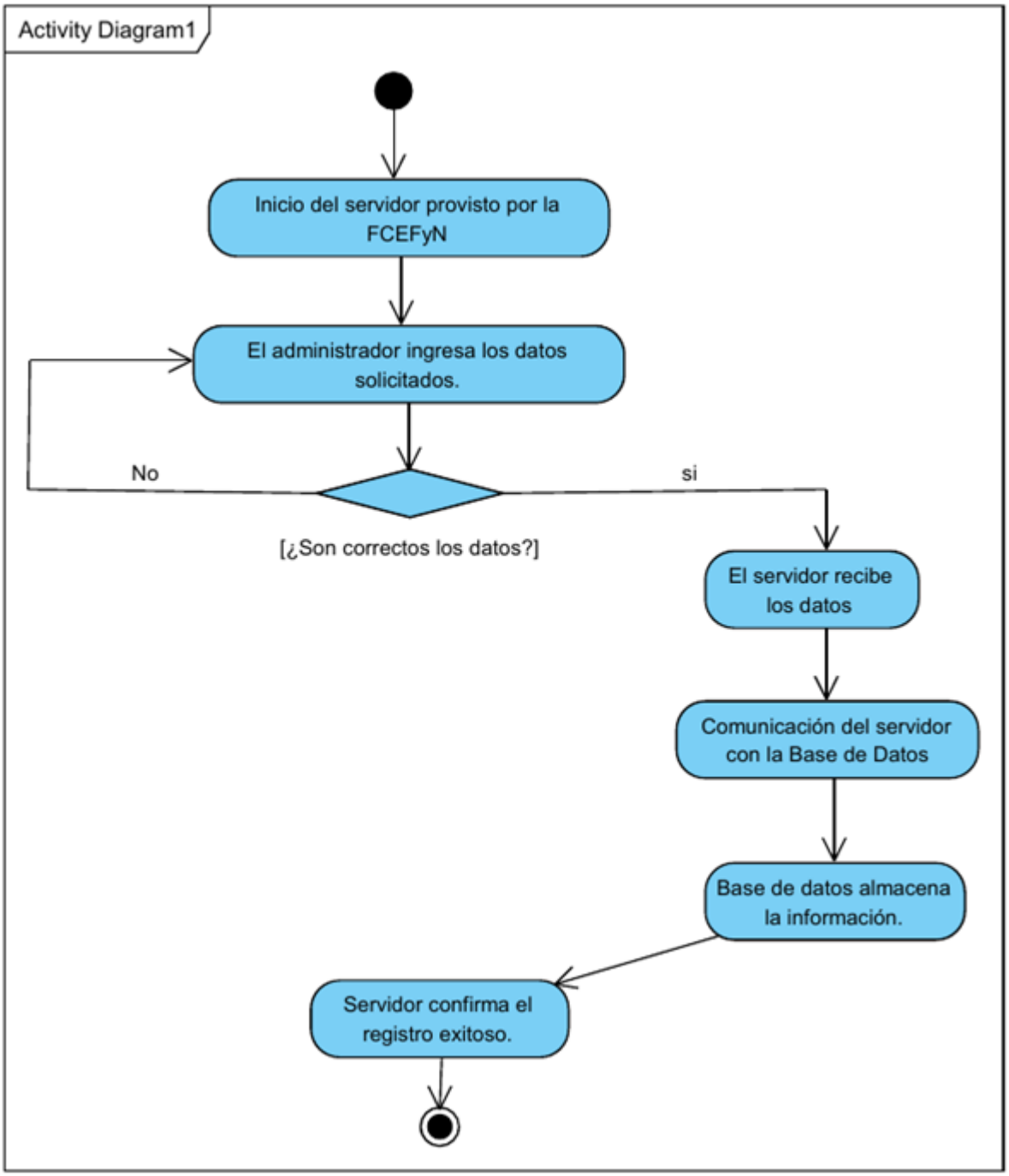
2. Vistas de Casos de Uso



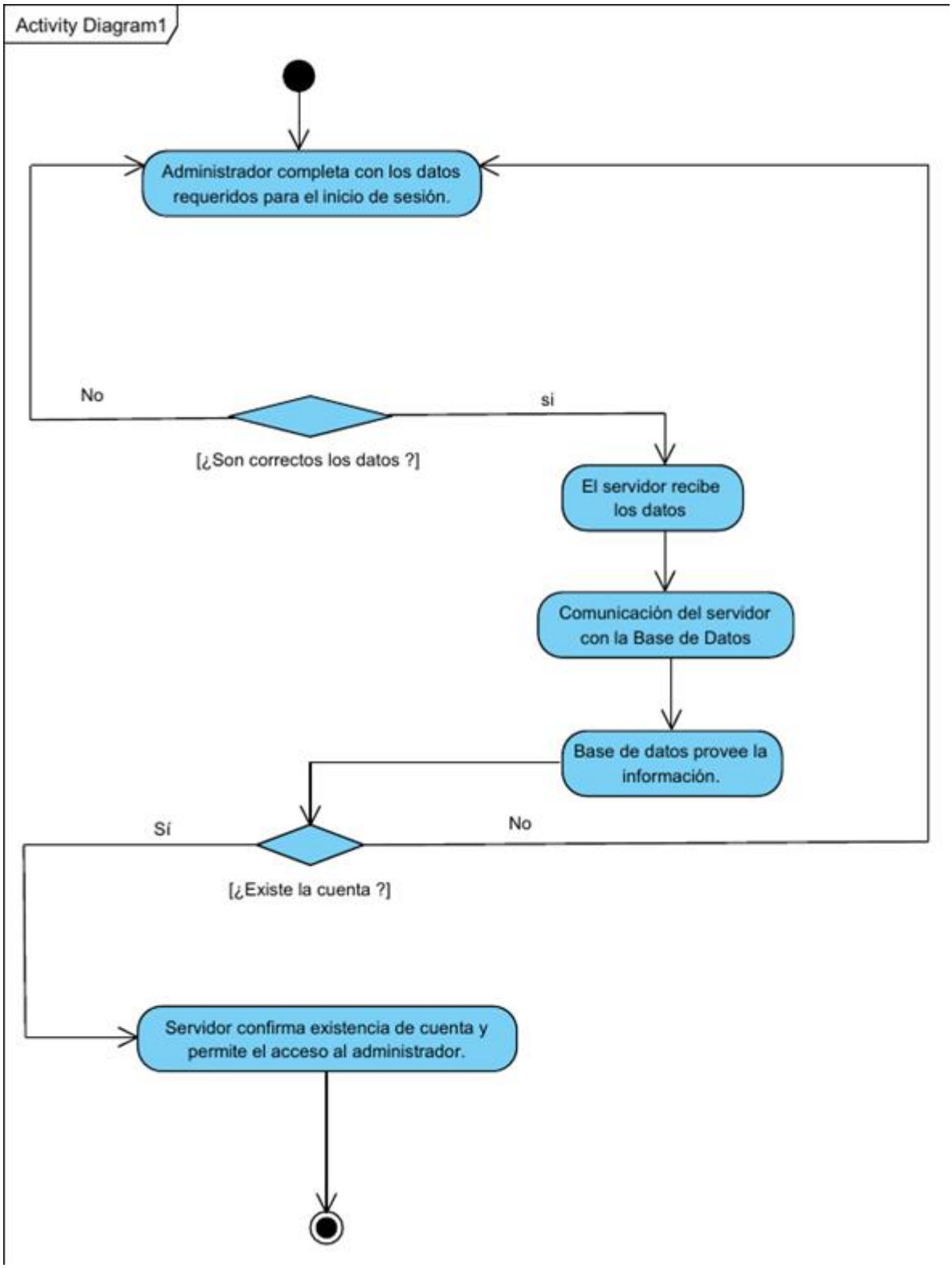
Casos de uso del diagrama anterior:

- CU1: Registrar cuenta de administrador.
- CU2: Realizar inicio de sesión
- CU3: Subir, modificar o eliminar publicación.
- CU4: Colocar formulario en publicación.
- CU5: Subir apuntes y demás documentos en la publicación.
- CU6: Ver publicaciones, noticias, avisos, datos.
- CU7: Descargar documentos y apuntes.
- CU8: Rellenar formularios y consultas.

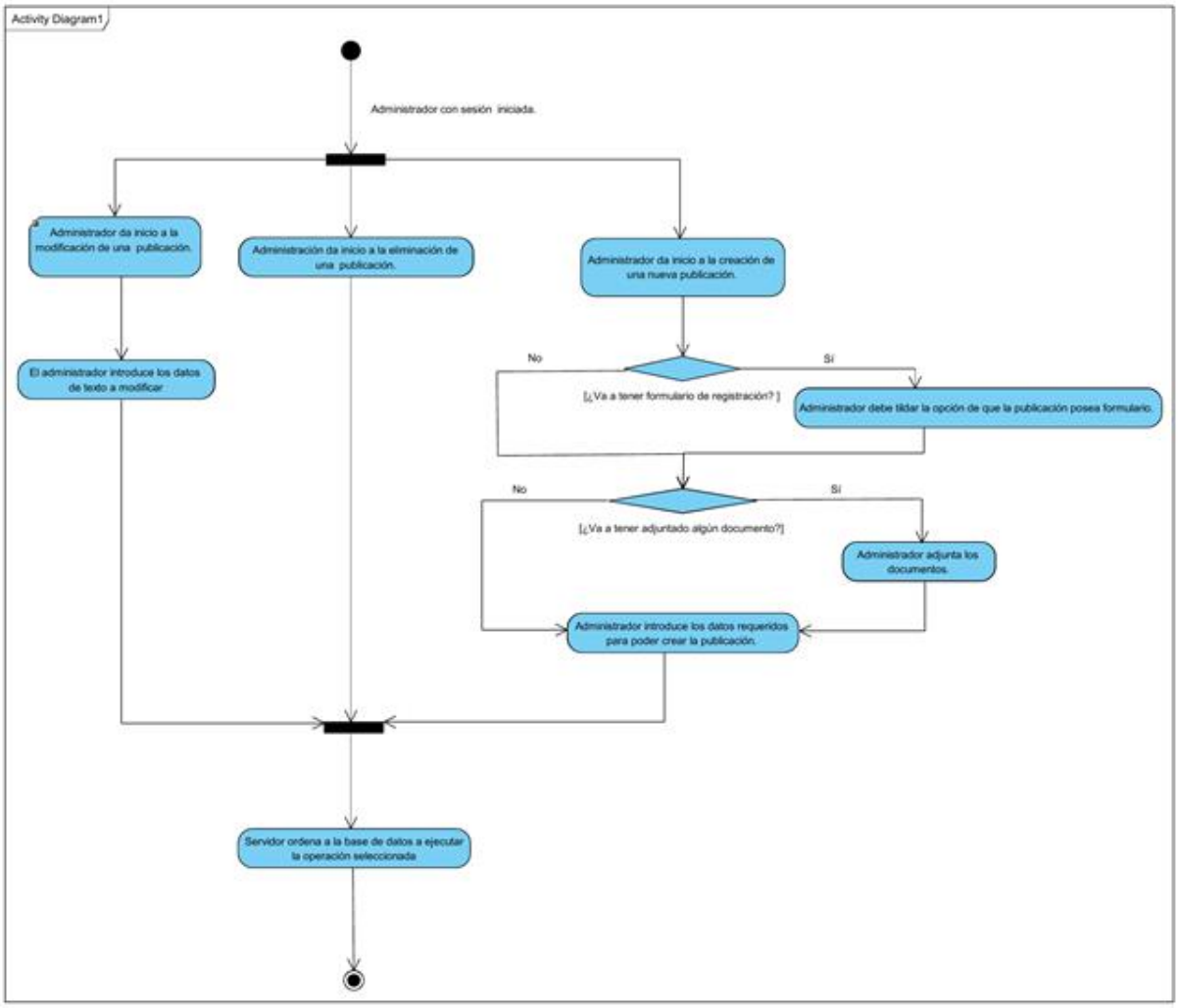
2.1. DIAGRAMA DE ACTIVIDAD CU1



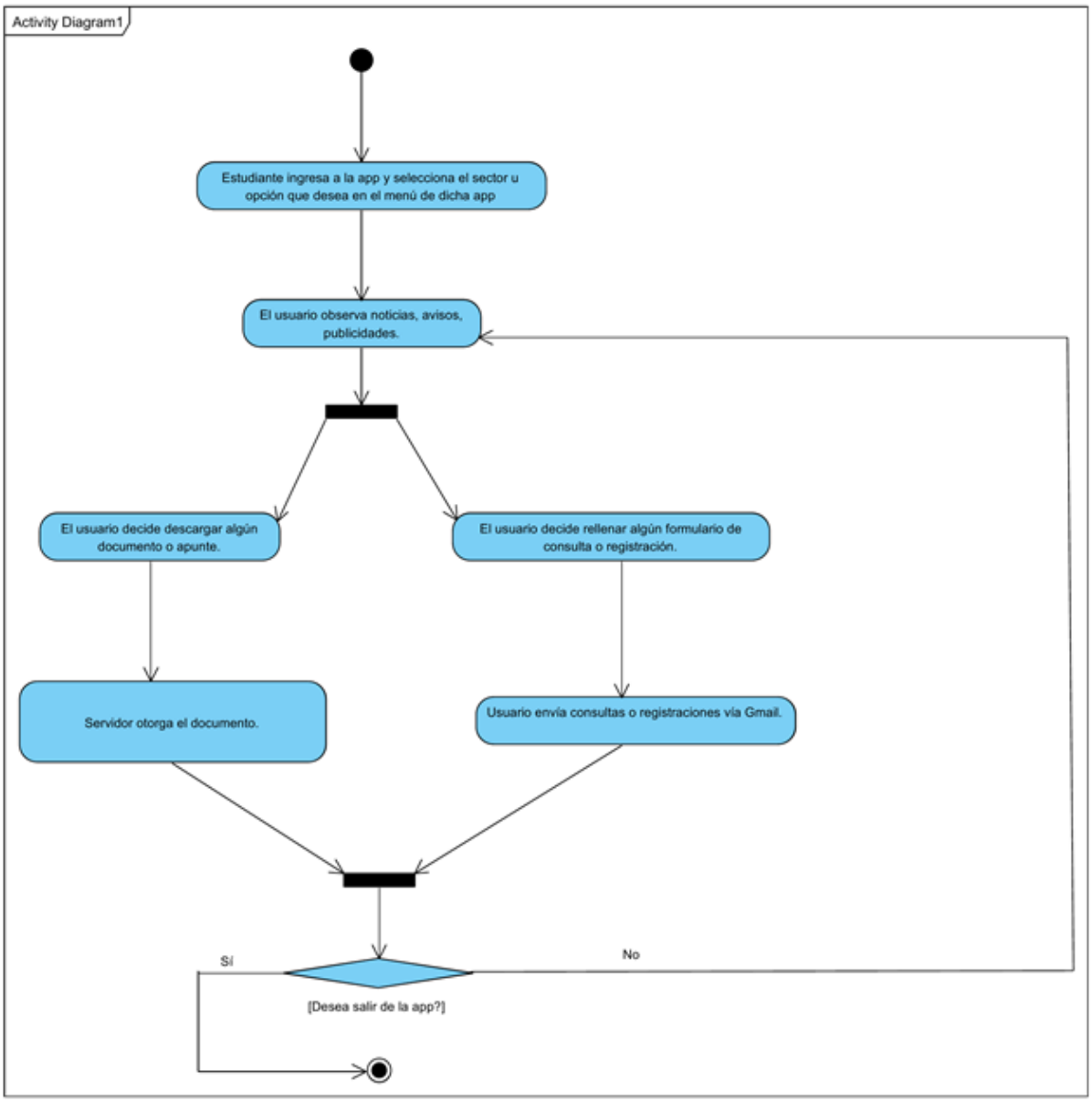
2.2. DIAGRAMA DE ACTIVIDAD CU2



2.3. DIAGRAMA DE ACTIVIDAD DE CU3, CU4 Y CU5

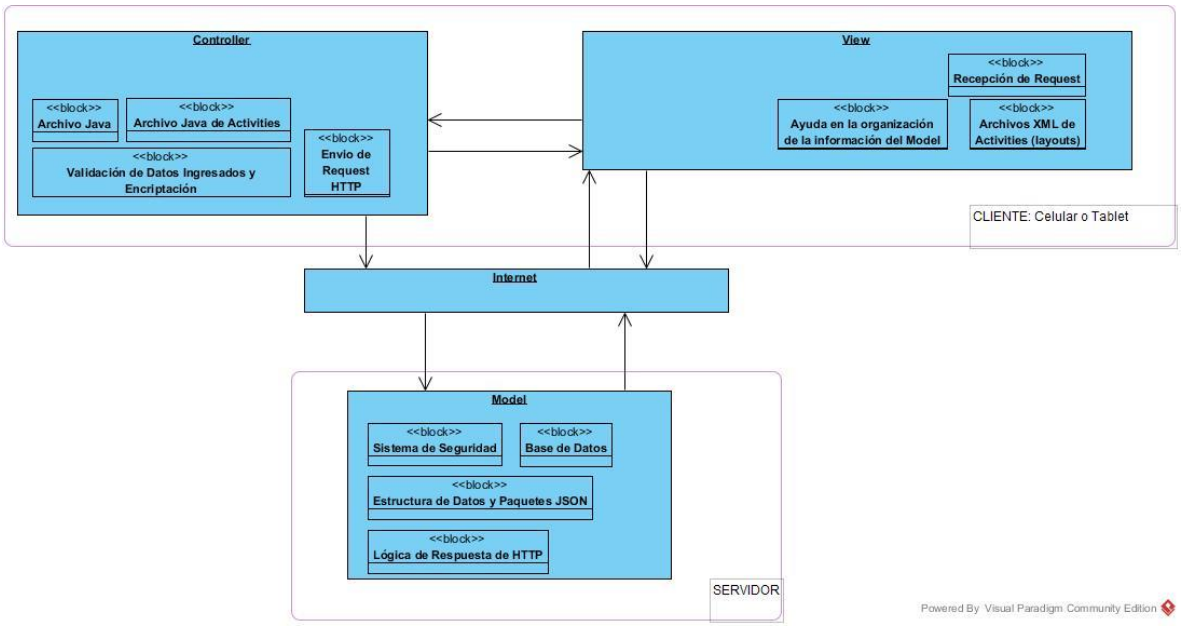


2.4. DIAGRAMA DE ACTIVIDAD DE CU6, CU7 Y CU8



3. DIAGRAMA DE ARQUITECTURA PRELIMINAR

Anotaciones preliminares



Módulos:

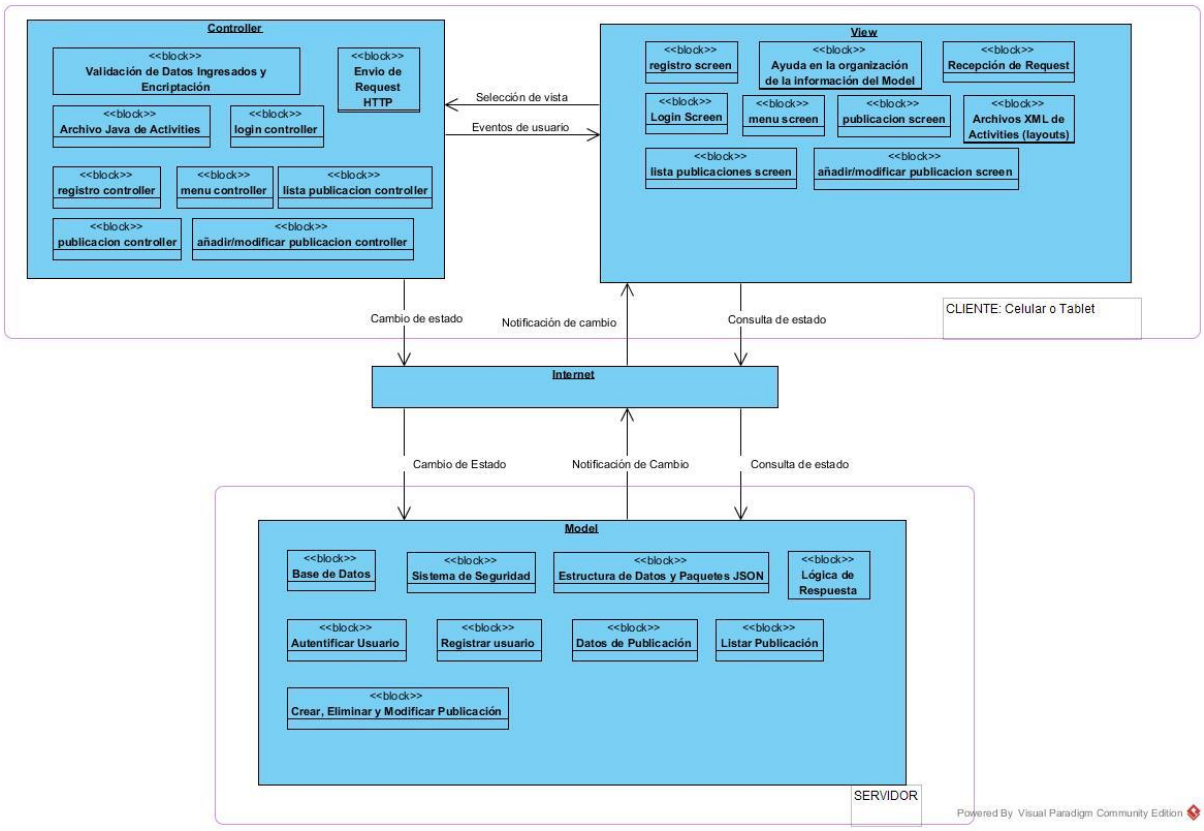
MVC:

View: Screen. (Login, registro, mostrar publicación, listar publicaciones, menú).

Controller: Peticiones Http y encriptado. Event Handler (1 controler por view). Actualización de views.

Model: Creación, modificación y eliminación de publicaciones. Administración de cuentas y seguridad. Notificación de actualización.

4. Descripción general del sistema.



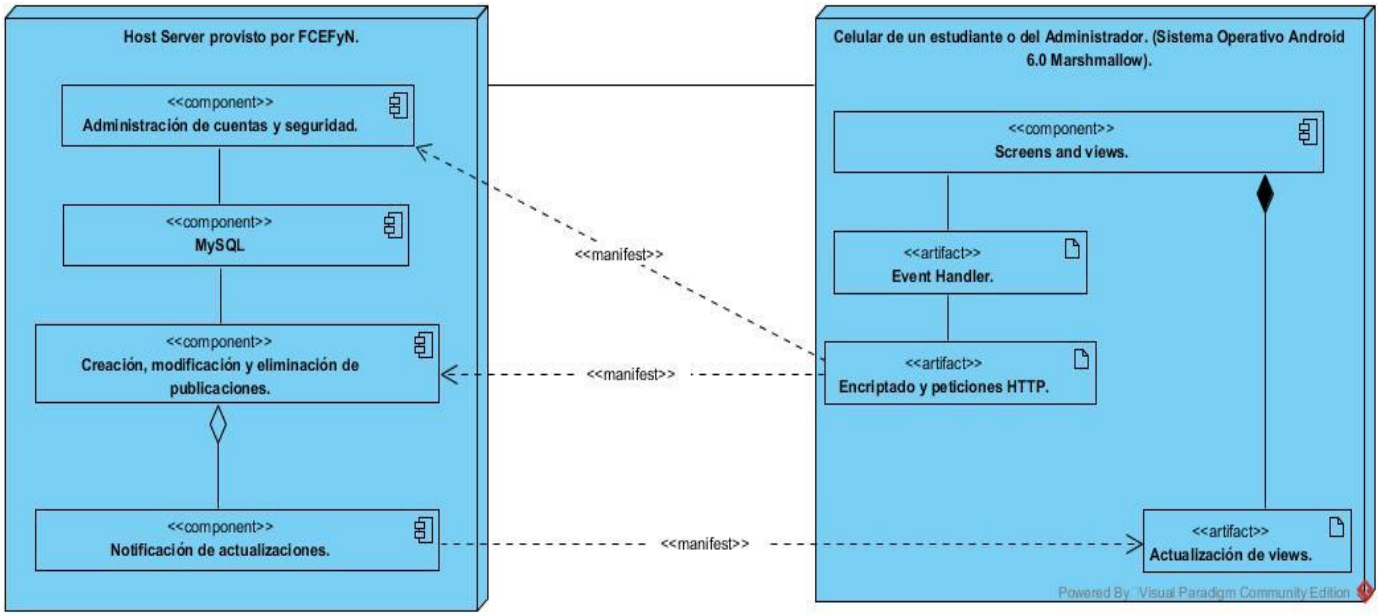
5. Definición de la Arquitectura

La arquitectura seleccionada para el desarrollo del sistema es una arquitectura MVC combinada con la arquitectura Cliente-Servidor.

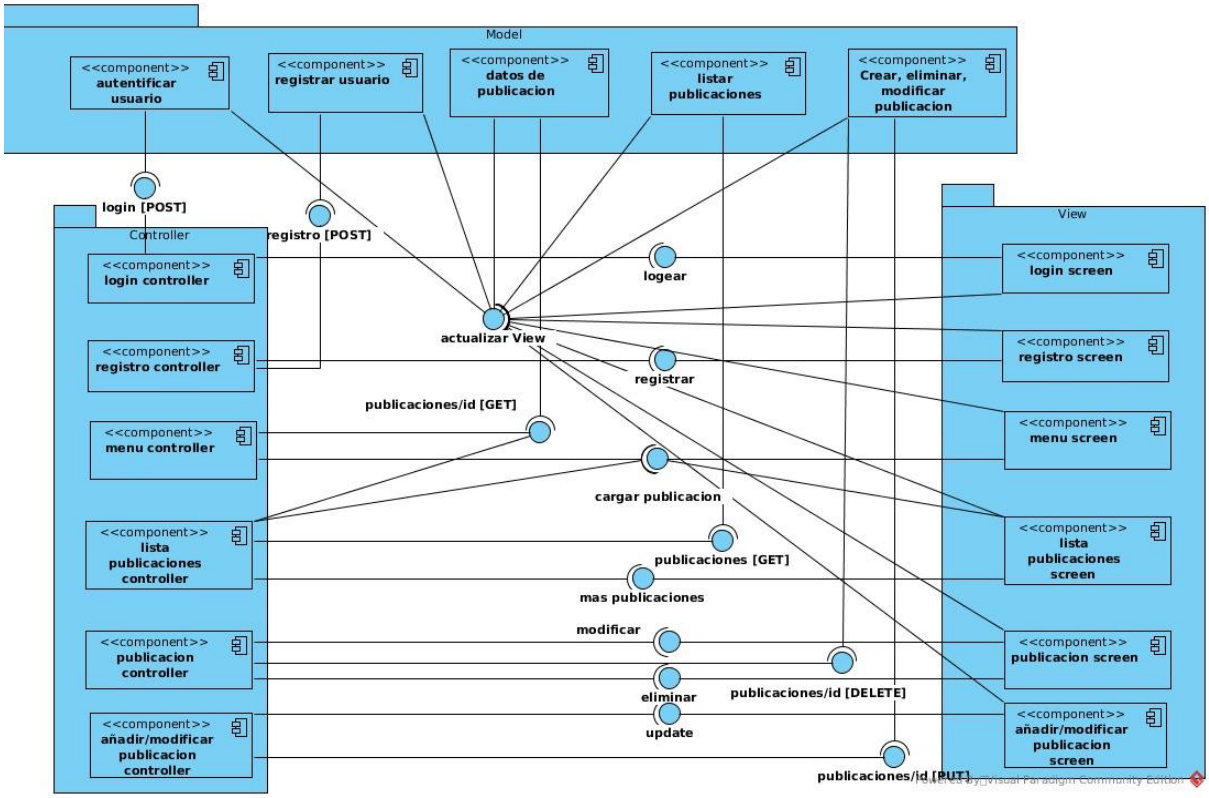
Elegimos MVC porque nos permite separar la *presentación* e *interacción de lo datos* del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Model maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista define y gestiona como se presentan los datos al usuario. El componente Controlador dirige la interacción del usuario y pasa estas interacciones a Vista y Modelo. Estos tres componentes se pueden visualizar en los diagramas de bloque anteriores. En el caso de la aplicación del presente trabajo se necesita tener diferentes interacciones y representación de la distintas noticias –o publicaciones- , y resulta beneficioso para esto aplicar el patrón MVC. Cabe destacar que debido a las formas de representación de interacción pueden cambiar en algún futuro, es muy probable que dicho patrón aporte grandes ventajas.

Además, podemos mencionar que utilizamos la Arquitectura Cliente-Servidor donde el Model es parte del servidor ofreciendo así mayor accesibilidad a través de la red.

6. Diagrama de despliegue.



7. Diagrama de componentes.



8. PRUEBAS DE INTEGRACIÓN

Consisten en pruebas similares a las pruebas unitarias del servidor, teniendo en cuenta para este caso que se posee una comunicación Cliente-Servidor, en donde existe una creación y recepción de paquetes JSON y request HTTP en ambas partes (tanto en el cliente como en el mencionado servidor). (Ver *Pruebas Unitarias. Documento de Diseño.*).