Modelo de Vistas de Arquitectura 4+1**- SafePet**

# **1. Introducción**

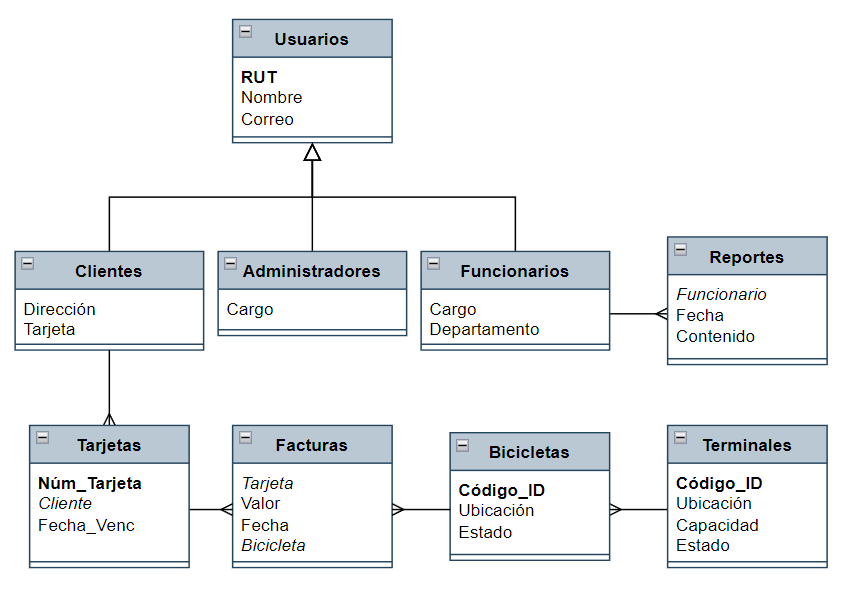
En este documento se registra la arquitectura del sistema utilizando el modelo de vistas 4+1. Cada una de las vistas del modelo tiene su propia sección, iniciando con la **vista lógica** en la sección 2, seguida de la **vista de procesos** en la sección 3, la **vista de desarrollo** en la sección 4, la **vista física** en la sección 5, y la **vista de escenarios** en la sección 6.

# 2. Vista Lógica

El modelo de la vista lógica encapsula la estructura estática del sistema en lo que respecta a su funcionamiento, y en particular, a los servicios, comportamientos y responsabilidades relacionadas a ello.

[ incluir un diagrama de clases UML o un diagrama entidad-relación ]

[ ejemplo vista logica ]

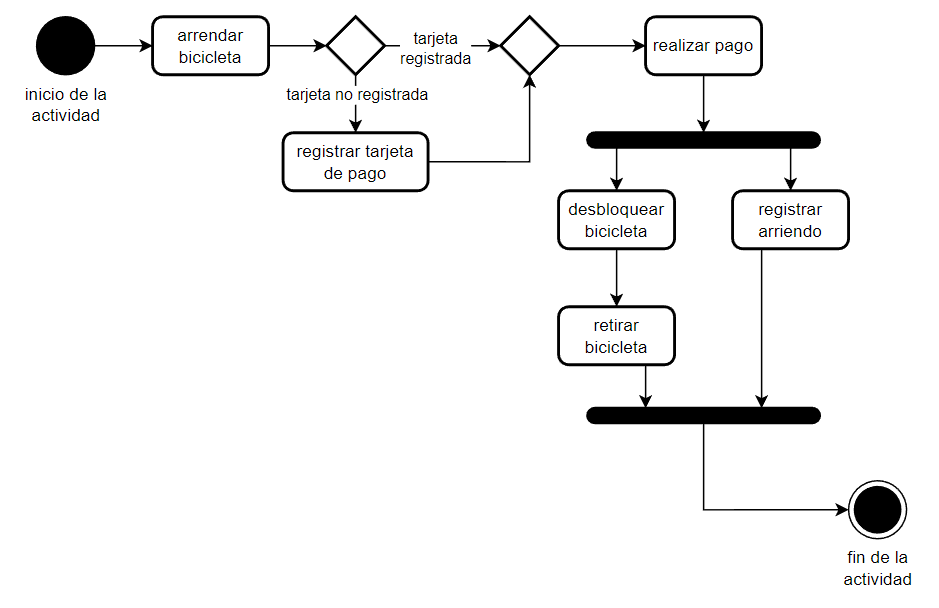
**

# 3. Vista de Procesos

El modelo de la vista de procesos encapsula la estructura dinámica del sistema, como propiedades no-funcionales del sistema; identifica los componentes y los procesos del sistema, y la forma en que interactúan o se ejecutan durante la ejecución del mismo, especialmente al hacerlo de forma concurrente.

[ incluir diagramas de actividad [2] o de secuencias [3] para cada proceso? ]

[ ejemplo de vista de procesos ]



# 4. Vista de Desarrollo

El modelo de la vista de desarrollo encapsula la estructura de la implementación del sistema, en lo que respecta a código fuente, incluyendo librerías y dependencias. Se enfoca en la naturaleza modular del software, y en las técnicas y herramientas utilizadas para el desarrollo del proyecto.

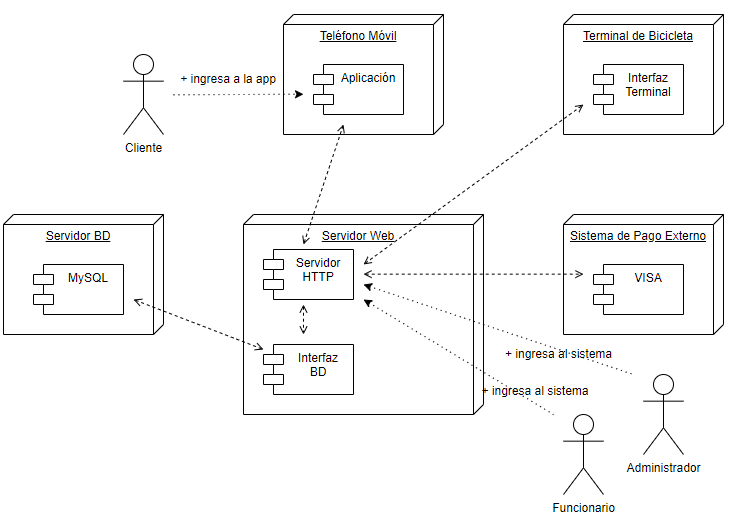
[ incluir ????? ]

# 5. Vista Física

El modelo de la vista física encapsula la distribución física del sistema ya en producción. Incluye la infraestructura de hardware utilizada.

[ incluir algun diagrama que indique la estructura física del sistema. onda el servidor web, los servidores de google, los de la ia que usemos, y el dispositivo movil de un usuario ]

[ ejemplo de vista fisica ]



# 6. Vista de Escenarios

La vista de escenarios permite verificar la validez de los modelos previos al diagramar casos de uso específicos, e indicando cómo los modelos permiten el correcto funcionamiento del sistema para completar dichos casos de uso.

[ incluir diagramas de actividad [2] o de secuencia [3], supongo ]

# 7. Bibliografía

1. “4+1”, Software Architecture Guild (original en inglés)

<https://software-architecture-guild.com/guide/competencies/modeling/frameworks/4_plus_1/>

1. “Diagrama de Actividad”, Wikipedia

<https://es.wikipedia.org/wiki/Diagrama_de_flujo>

1. “Diagrama de Secuencia”, Wikipedia

<https://es.wikipedia.org/wiki/Diagrama_de_secuencia>