

**INFORME:**

**ENCARGO INGENIERÍA DE SOFTWARE - EVALUACIÓN PARCIAL 2 “MASTERBIKES”**

Alumnos: Nicolás Bello, Anton Knittel, Rodrigo Vargas, Francisco Vera

Profesor: José Luis Silva

Asignatura: Ingeniería de Software

Caso de estudio: Bicicletas MasterBikes

Tercer Semestre, Segundo Año

20 de Mayo de 2025

| **Equipo de Desarrollo** | |
| --- | --- |
| **Nombre** | **Rol** |
| Rodrigo Vargas | Gestor del Proyecto |
| Nicolás Bello | Analista |
| Francisco Vera | Diseñador |
| Anton Knittel | Tester |

**TABLA DE CONTENIDO**

[**INTRODUCCIÓN 4**](#_9ikrqmvp7p23)

[**DIAGRAMAS DE VISTA 4+1 5**](#_10ld2yo31qix)

[1. DIAGRAMA DE ESTADO 5](#_4mz2a73jnuq8)

[1.1. ARRIENDO DE UNA BICICLETA 6](#_8f5yvsrj1e6k)

[1.2. VENTA DE UNA BICICLETA 6](#_2ys0mey2ut8)

[1.3. REPARACIÓN DE UNA BICICLETA 7](#_t6qqzahv09t0)

[1.4. DESPACHO DE UNA BICICLETA 7](#_keghknq2mmce)

[2. DIAGRAMA DE SECUENCIA 8](#_wm1kl5sokl0u)

[2.1. REPARACIÓN DE UNA BICICLETA 8](#_tqzyudtyi62)

[3. DIAGRAMA DE CASOS DE USO 9](#_qtvo5hwp1na2)

[3.1. COMPRA-VENTA DE UNA BICICLETA 9](#_xtvm4g6tpp4i)

[3.2. REPARACIÓN DE UNA BICICLETA 10](#_km5hu09l68b6)

[4. DIAGRAMA DE COMPONENTES 11](#_2a0esqfl2w7o)

[5. DIAGRAMA DE DESPLIEGUE 11](#_rt6wf2q0l8w2)

[6. DIAGRAMA DE CLASES 12](#_xgknqcezl5rg)

[**CALIDAD EN EL DISEÑO ARQUITECTÓNICO DEL SOFTWARE 13**](#_2bv1j6wsbnb5)

[7. ESTÁNDARES DE CALIDAD APLICADOS EN EL DISEÑO DE LA ARQUITECTURA DEL SOFTWARE 13](#_njktbc2yzxmv)

[7.1. MODULARIDAD 13](#_78t3v63m3kb6)

[7.2. ESCALABILIDAD 13](#_hzjj1q22b2ut)

[7.3. MANTENIBILIDAD 13](#_27v8vvxr5nrv)

[7.4. FIABILIDAD 13](#_5macjdvinht5)

[7.5. USABILIDAD 13](#_w1fj3hg1swc2)

[8. HERRAMIENTAS Y ESTÁNDARES APLICADOS: 13](#_n9d07wc2dwhk)

[9. JUSTIFICACIÓN Y BENEFICIOS 14](#_2xb48fdxt0uq)

[**CONCLUSIÓN 15**](#_7okhf5b30g6d)

# 

# **INTRODUCCIÓN**

En el marco del proceso de modernización de la empresa MasterBikes, se ha desarrollado una serie de diagramas siguiendo el modelo de vistas 4+1, con el fin de representar de forma clara la arquitectura del nuevo sistema. Este modelo permite abordar el diseño desde diferentes perspectivas clave para su desarrollo y comprensión. Los diagramas elaborados incluyen:

1. Casos de uso
2. Secuencia
3. Estado
4. Componentes
5. Despliegue

Estos diagramas permiten comprender cómo funcionará la nueva plataforma de MasterBikes, orientada a mejorar la atención al cliente y ampliar sus servicios a arriendo, reparación y seguimiento de productos.

# **DIAGRAMAS DE VISTA 4+1**

El diagrama de vistas 4+1 define diferentes perspectivas para describir la arquitectura del sistema mediante diagramas UML el cual es un estándar gráfico que permite representar y documentar diferentes aspectos de un sistema de software, facilitando su diseño y comunicación entre el equipo. En este sentido el modelo 4+1 organiza la arquitectura del sistema en cinco vistas:

* Lógica: estructura funcional (diagramas de clases).
* Procesos: comportamiento dinámico y concurrencia.
* Desarrollo: organización del código y módulos.
* Física: distribución en hardware.
* Casos de Uso (+1): escenarios reales que integran las vistas.

El uso conjunto del modelo 4+1 y UML en MasterBikes permitirá construir una arquitectura clara, modular y adaptable, que soporta la transformación digital de la empresa y la integración de nuevos servicios, contribuyendo a la calidad, mantenibilidad y escalabilidad del sistema.

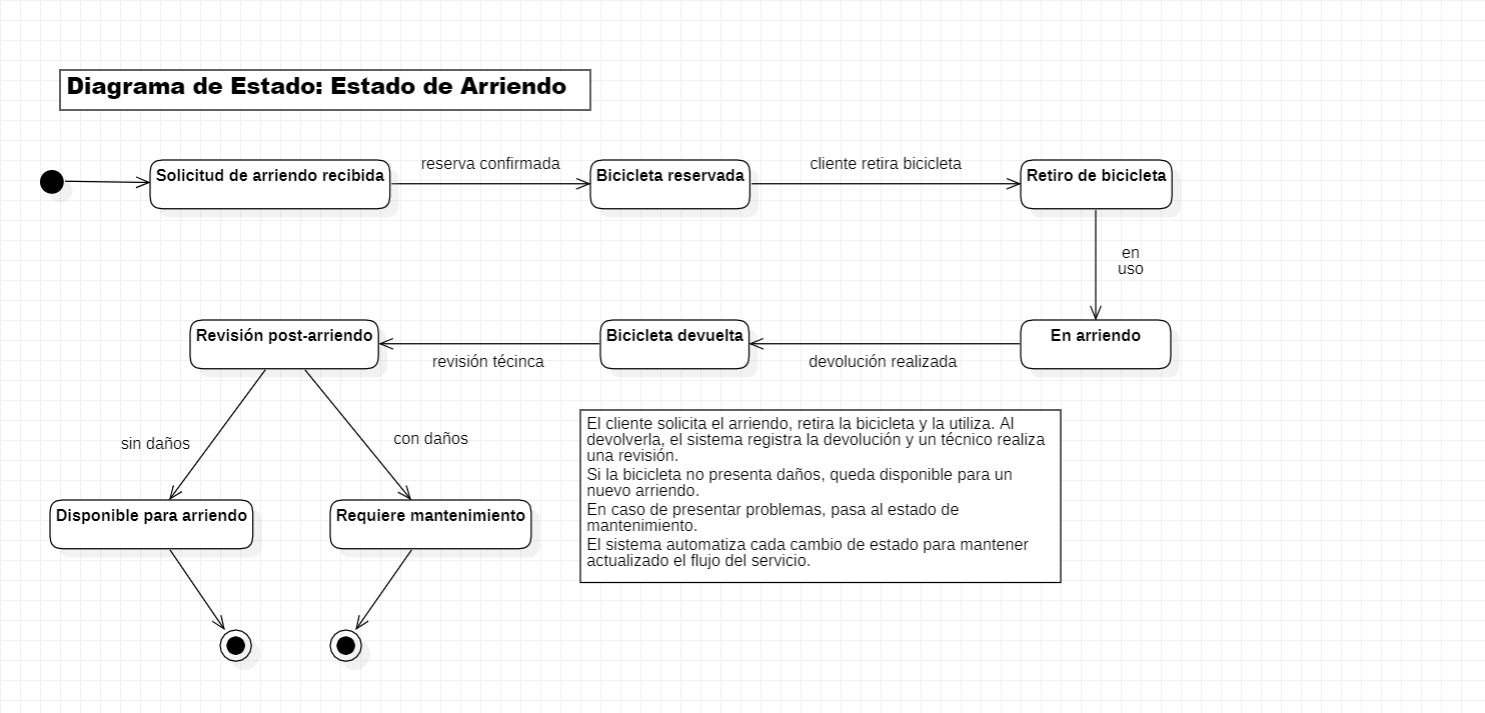
## **DIAGRAMA DE ESTADO**

Un diagrama de estado es un tipo de diagrama UML que se utiliza para describir el comportamiento de un objeto a lo largo del tiempo en respuesta a eventos (Acciones o hechos que ocurren en el sistema), mostrando los diferentes estados por los que pasa y cómo cambia de estado debido a estos eventos,acciones o condiciones específicas.

Los diagramas de estado se utilizaran para modelar los cambios de estado de las bicicletas a lo largo de distintos procesos (arriendo, venta, reparación, despacho). Permitiendo visualizar claramente los eventos que afectan al ciclo de vida de los objetos.

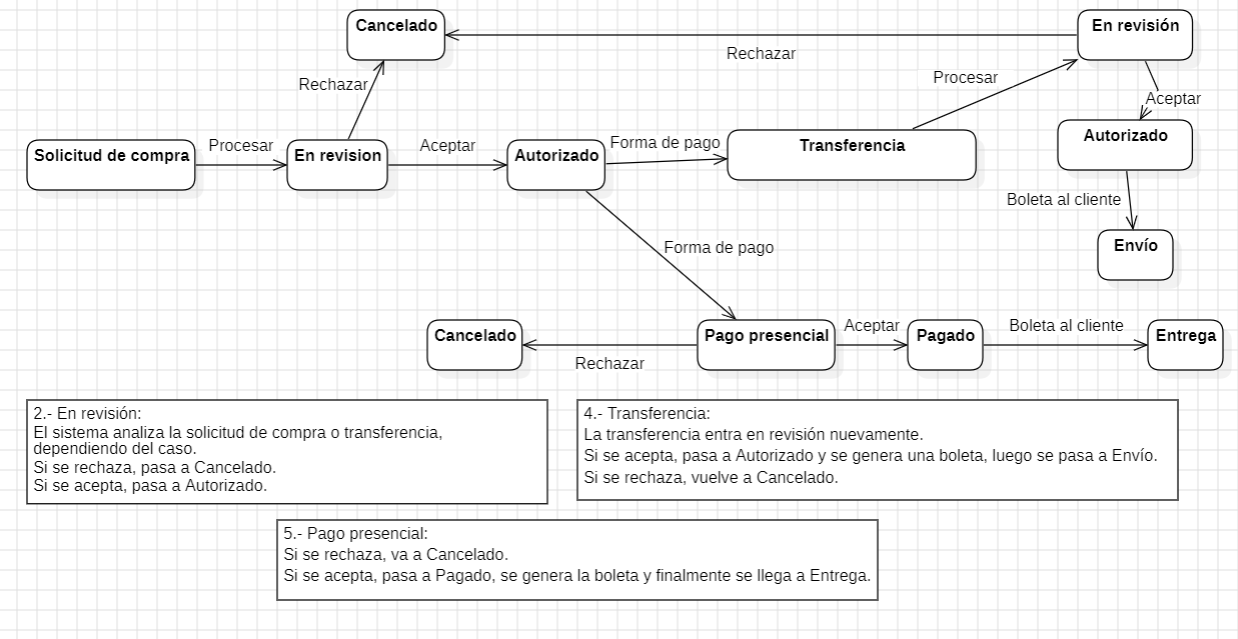
Beneficios: Favorecen el entendimiento del comportamiento interno del sistema, útil para testing y validación.

### ARRIENDO DE UNA BICICLETA

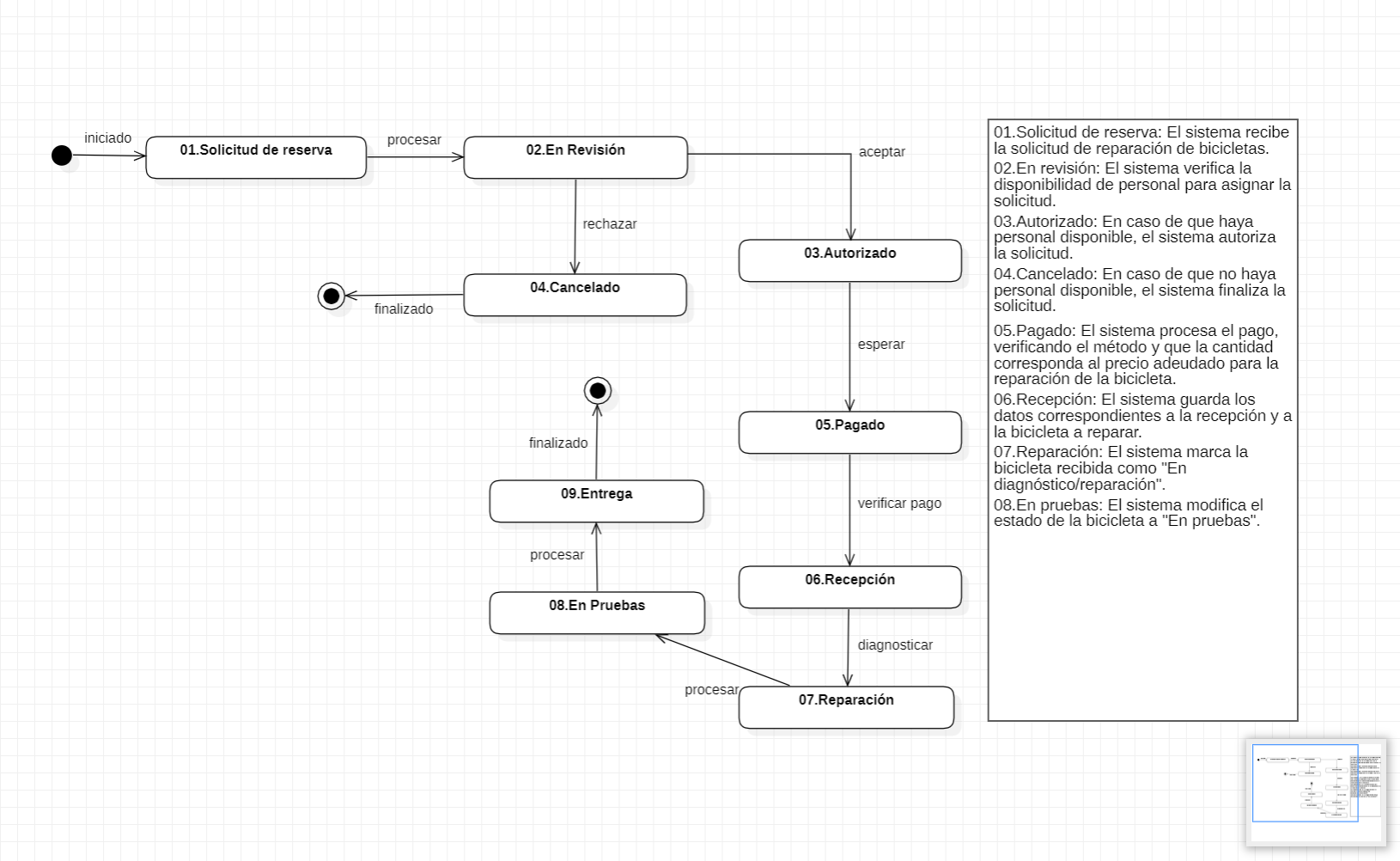


### 

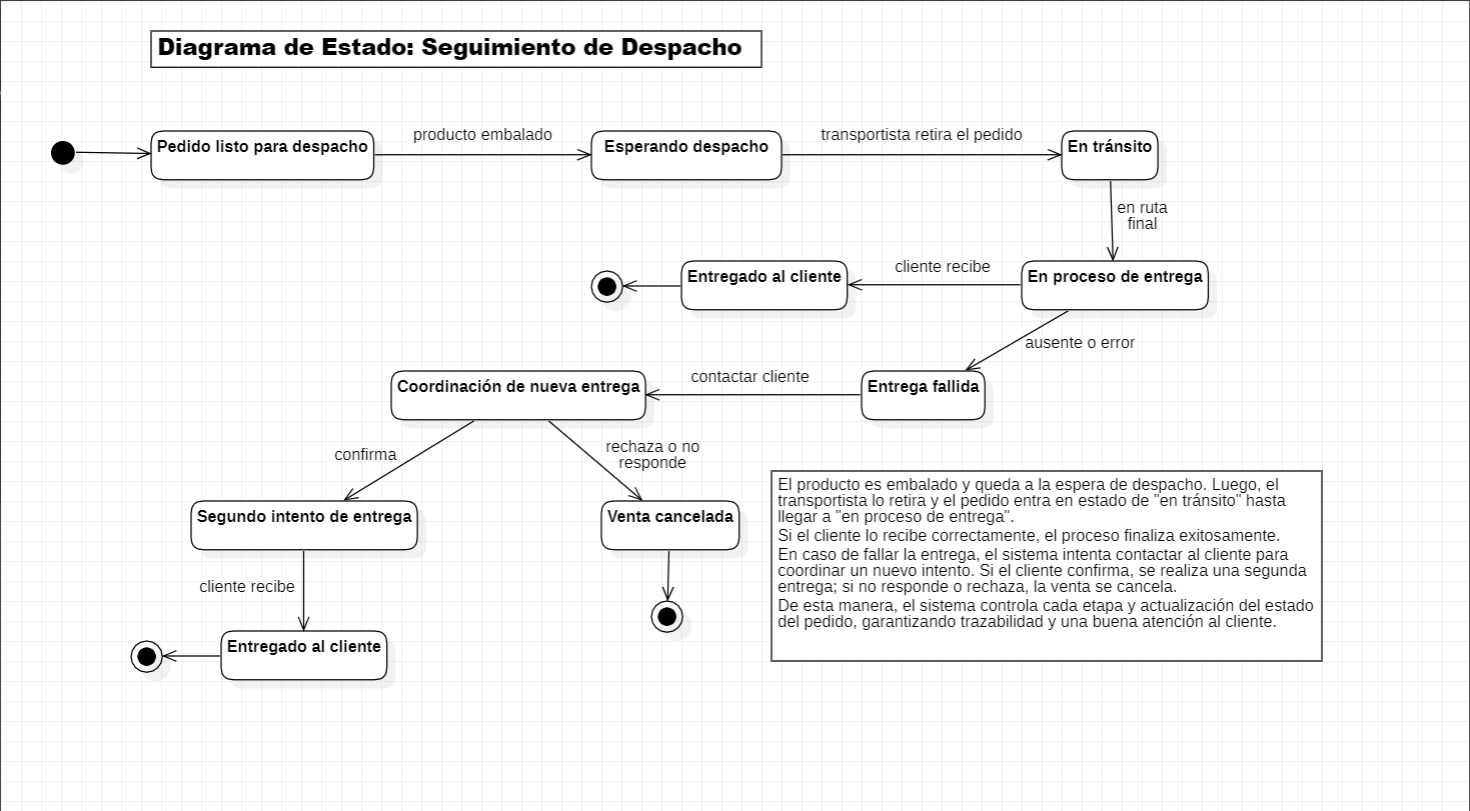
### VENTA DE UNA BICICLETA

****

### REPARACIÓN DE UNA BICICLETA

****

### DESPACHO DE UNA BICICLETA



## 

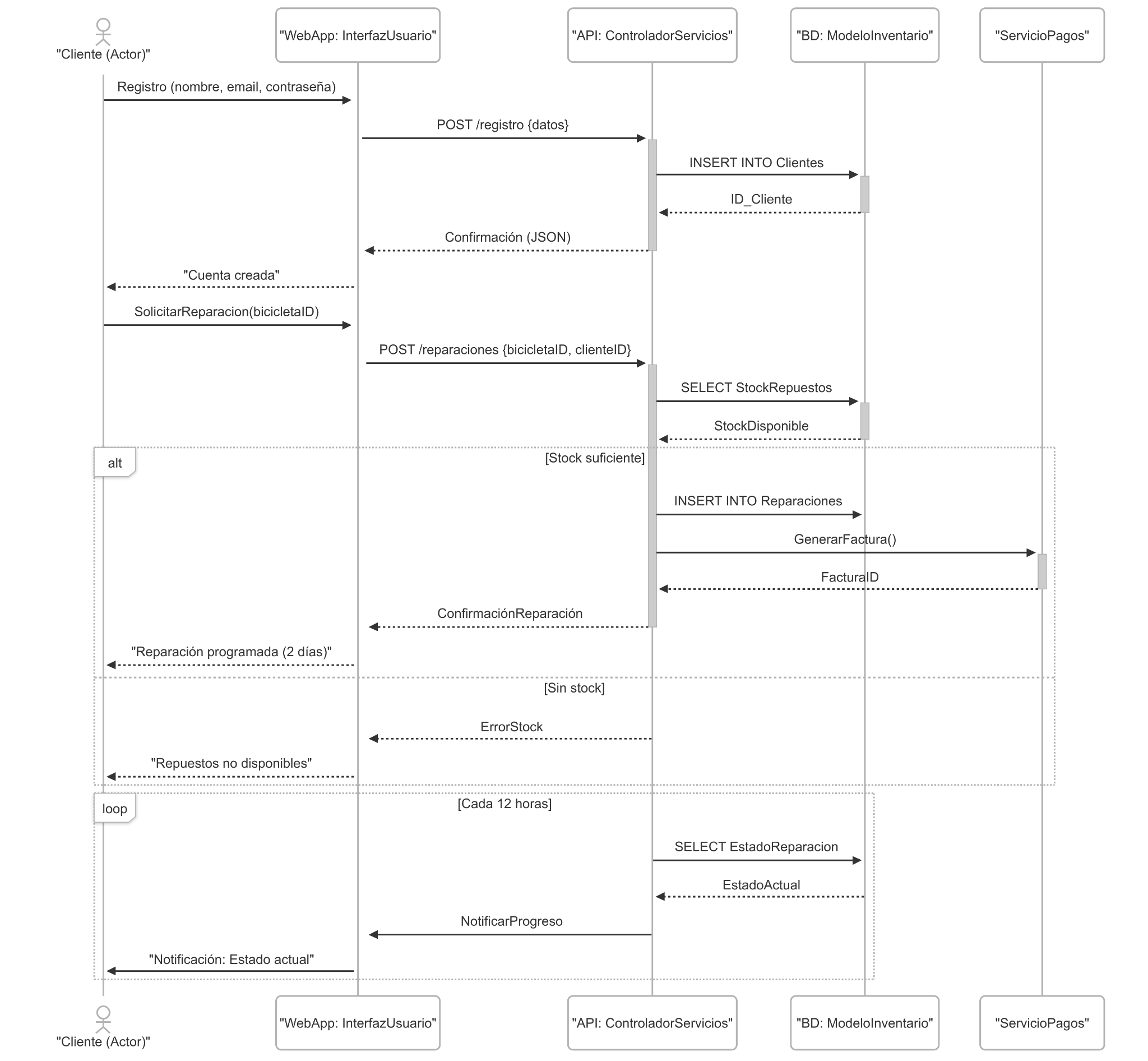
## 

## **DIAGRAMA DE SECUENCIA**

Un diagrama de secuencia es un tipo de diagrama de la UML, que se utiliza para modelar la interacción entre objetos o actores en un sistema, mostrando el orden en el que ocurren los mensajes o eventos a lo largo del tiempo (Por ejemplo, durante la reparación de una bicicleta).

Beneficios: Ayudar a identificar el flujo de mensajes, detectar cuellos de botella y diseñar procesos eficientes.

### REPARACIÓN DE UNA BICICLETA

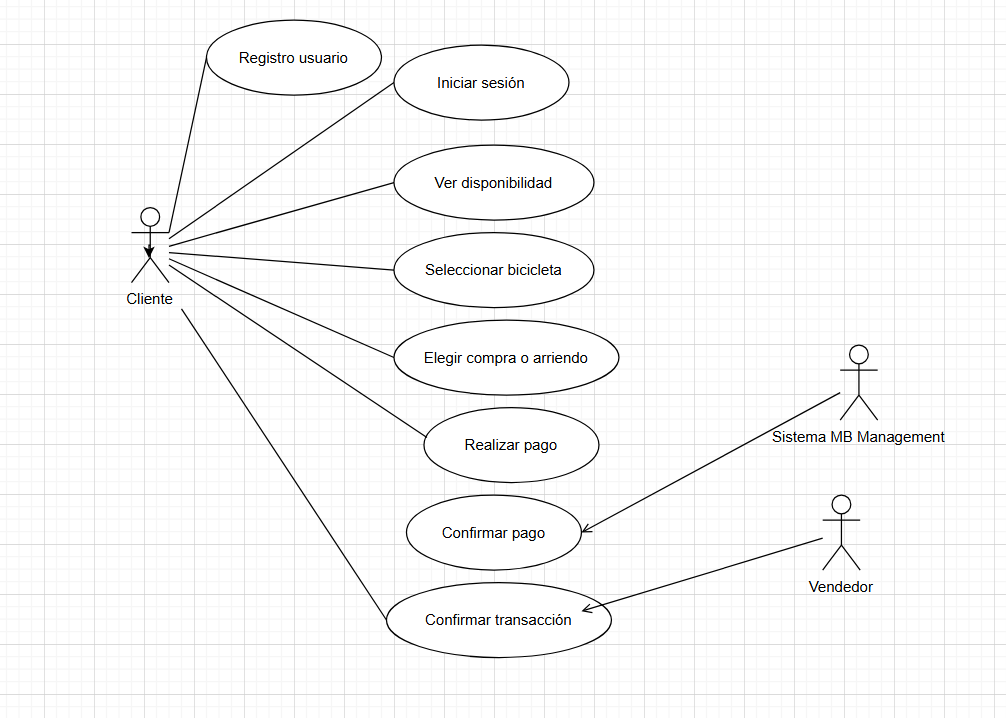


## **DIAGRAMA DE CASOS DE USO**

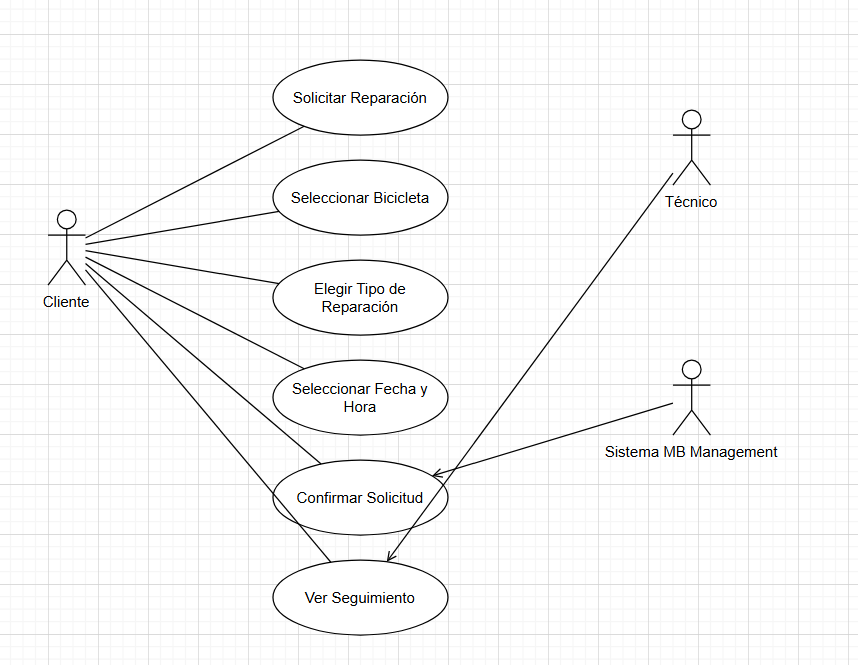
Un diagrama de casos de uso es una técnica utilizada en el análisis y diseño de sistemas para describir las interacciones entre los actores (usuarios u otros involucrados) y el sistema en desarrollo. Su objetivo es representar los requerimientos funcionales del sistema, mostrando de forma clara qué funcionalidades ofrece y cómo los usuarios interactúan con ellas (ej. compra, reparación de bicicletas).

Beneficios: Facilitar la recolección y validación de requerimientos con stakeholders.

### COMPRA-VENTA DE UNA BICICLETA



### REPARACIÓN DE UNA BICICLETA

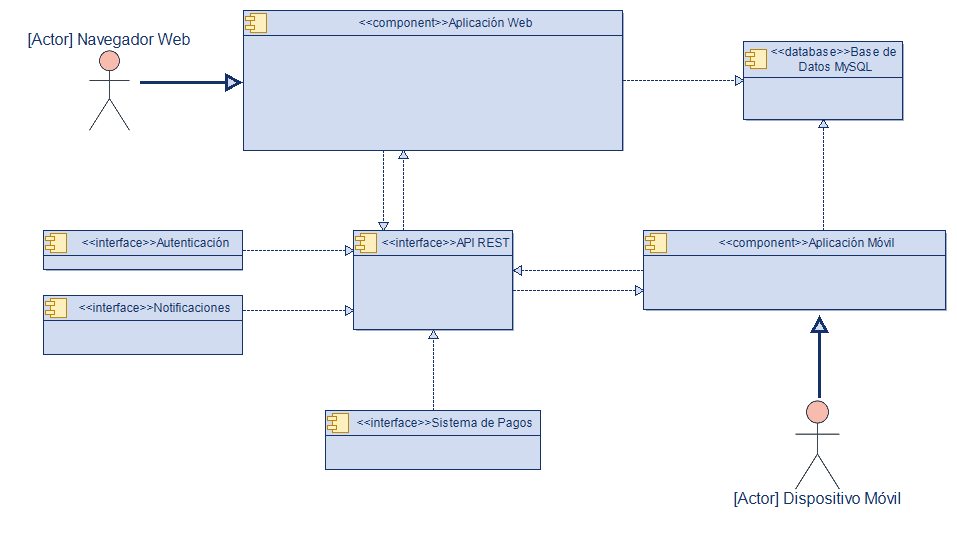


## 

## **DIAGRAMA DE COMPONENTES**

Un diagrama de componentes es un tipo de diagrama UML que muestra cómo está estructurado un sistema en componentes físicos o lógicos y cómo se relacionan entre ellos.

Beneficios: Mejorar la mantenibilidad, facilitar el trabajo en equipo y soportar la reutilización de componentes.



## **DIAGRAMA DE DESPLIEGUE**

Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura física del sistema, es decir, cómo los componentes del software se distribuyen y ejecutan en la infraestructura de hardware (servidores, dispositivos, nodos, etc.).

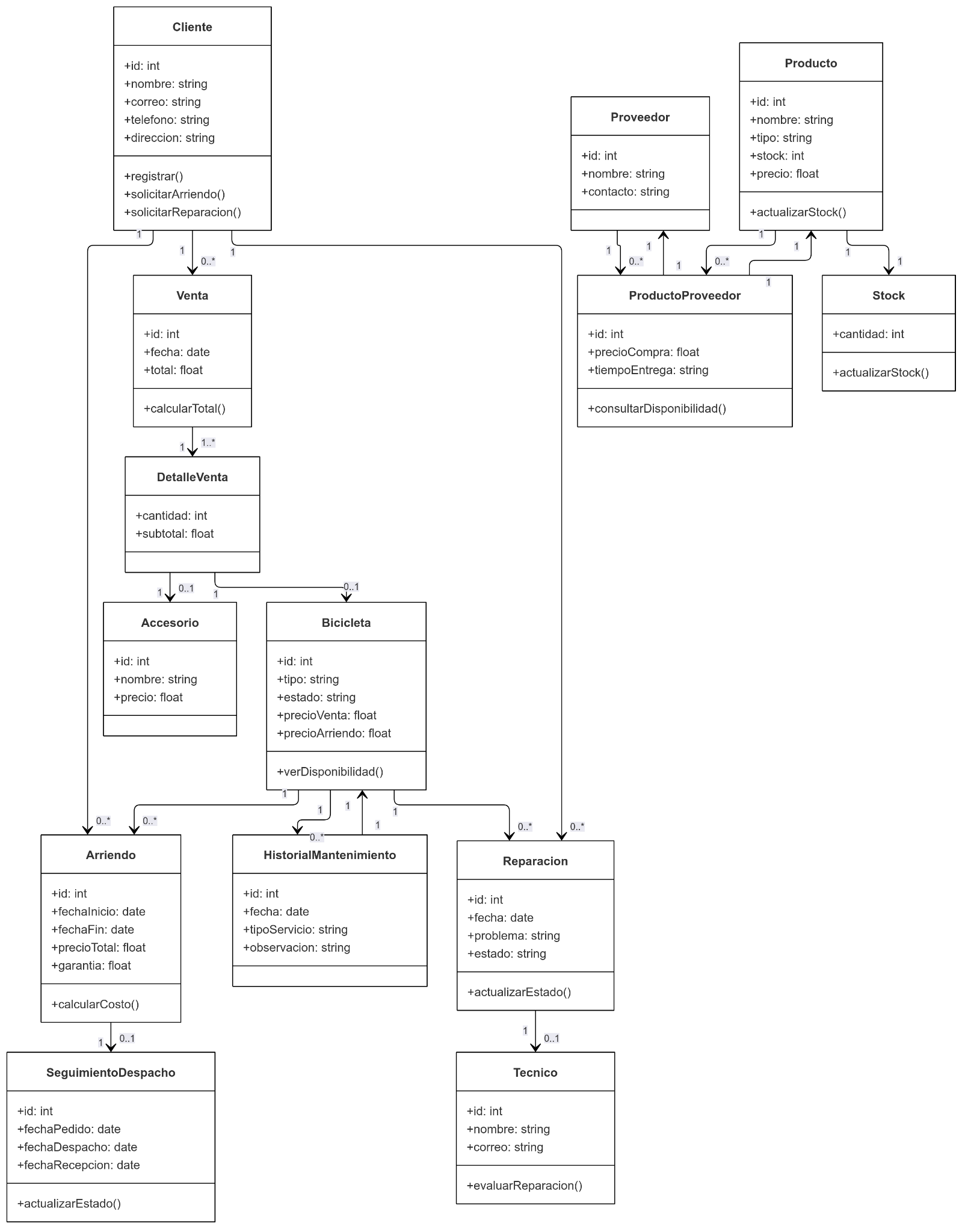
Beneficios: Permite planificar la infraestructura, optimizar rendimiento y asegurar escalabilidad.

## 

## **DIAGRAMA DE CLASES**

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Nos sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de convencimiento.

Beneficios: Nos ayuda a representar la estructura del sistema para una mejor comprensión del mismo. Cómo se relacionan las clases entre sí, qué atributos comparten, sirve de base para la implementación, además de permitir identificar errores conceptuales de manera temprana.



# **CALIDAD EN EL DISEÑO ARQUITECTÓNICO DEL SOFTWARE**

## **ESTÁNDARES DE CALIDAD APLICADOS EN EL DISEÑO DE LA ARQUITECTURA DEL SOFTWARE**

### MODULARIDAD

Separación clara de responsabilidades por componente.

### ESCALABILIDAD

Capacidad del sistema de crecer sin rediseño completo.

### MANTENIBILIDAD

Facilidad para modificar el sistema.

### FIABILIDAD

Comportamiento estable ante errores.

### USABILIDAD

Facilidad de uso para los usuarios finales.

## 

## **HERRAMIENTAS Y ESTÁNDARES APLICADOS:**

* UML (Unified Modeling Language)

Lenguaje visual estandarizado para representar el diseño y la estructura de un sistema de software, esto realizado a través de diagramas UML.

* Modelo de vistas 4+1 de Kruchten

Metodología para estructurar la arquitectura desde múltiples perspectivas.

* IEEE 1016-2009

Estándar para la descripción de arquitecturas de software.

## 

## **JUSTIFICACIÓN Y BENEFICIOS**

* UML

Permite una comunicación clara y estandarizada entre los miembros del equipo.

* Modelo de vistas 4+1

Asegura que se cubren aspectos funcionales, estructurales, de comportamiento y despliegue, alineando diseño con requisitos.

* IEEE 1016-2009

Estándar que ayuda a organizar y documentar bien la arquitectura del software, dejando claro cómo está construido el sistema. Esto permite que cualquier persona del equipo pueda entenderlo fácilmente, hacer mejoras o solucionar problemas en el futuro, y tomar decisiones con mejor información.

# 

# **CONCLUSIÓN**

Con la realización del modelo de vistas 4+1 damos por concluida esta primera parte de la etapa de diseño dentro del ciclo de vida del software. Gracias a cada uno de los diagramas desarrollados hemos podido comprender de mejor manera que hará cada parte del sistema, como se comportará y cómo se comunicará con los diferentes actores que harán uso de las herramientas. Por otro lado, y no por ello menos importante, esperamos que la realización de estos diagramas sea de ayuda para los diferentes stakeholders, quienes cuentan de manera individual con visiones diferentes sobre el proyecto. Finalmente, comprendemos que haber llevado a cabo esta labor nos brindará de múltiples ventajas hacia las fases posteriores del proyecto, tales como una mejor organización como equipo de trabajo, facilidad de documentación dada la división de la arquitectura en varias vistas, entre otras, como el hacer que nos enfoquemos en alinear el sistema con los casos de uso.