

**Universidad de Guadalajara**  
**C.U.C.E.I.**  
**Sem. de ingeniería de software 1**  
**Sección: Do1**  
**Mtra.: Karla Avila Cárdenas**



**Práctica 9:**  
**Diseño lógico-físico**  
**Shimoda Emily Tomomi**  
**216579882 - Ingeniería en Computación**  
**Luis Daniel Zamora Delgadillo**  
**217761153 - Ingeniería Informática**

**21B**

## Índice

Índice	2
Introducción	3
Desarrollo	5
Diagrama de estado	5
Diagramas de actividades	6
Diagrama de componentes	20
Resultados	21
Diagrama de despliegue	21
Conclusiones	21
Referencias	22

## Introducción

Los diagramas que se realizarán en esta actividad serán los siguientes:

- **Diagrama de estado:** Aunque este diagrama solo consta de unos pocos elementos, si se utiliza correctamente, puede contribuir notablemente al resultado final. Un diagrama de estado UML (también llamado diagrama de estado, diagrama de transición de estados o diagrama de máquina de estados) muestra los estados por los que pasa una máquina de estados finitos, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. El diagrama proporciona un estado inicial y uno final, así como al menos un estado intermedio para cada objeto. El diagrama de estado permite, de este modo, representar el ciclo de vida completo de cualquier sistema, subsistema o componentes o clases del mismo

El objetivo de los diagramas de estado es describir el comportamiento de un sistema con la máxima precisión. Entre otras cosas, esta representación gráfica de los procesos debería dar respuesta a las siguientes preguntas:

- ¿Qué sucede cuando el objeto está en un estado concreto?
- ¿En qué estado debe estar el objeto para cambiar de comportamiento?
- ¿Cuáles son los desencadenantes?
- ¿Qué propiedades debe tener el objeto para poder cambiar de estado?

- **Diagrama de actividades:** El diagrama de actividades es un tipo de diagrama dentro del lenguaje unificado de modelado (UML). Este lenguaje de modelado gráfico define formas para la representación de la programación orientada a objetos

Los diagramas de actividades UML pertenecen al grupo de diagramas de comportamiento en UML. Mientras que un diagrama de estructura registra el estado de un sistema, es decir, los objetos existentes y sus jerarquías, así como las conexiones entre ellos en un momento determinado, los diagramas de comportamiento describen el flujo cronológico de la circulación de datos. Además del diagrama de actividades, el diagrama de caso de uso y el diagrama de máquina de estados también pertenecen a este grupo. Los diagramas de actividades son similares en su uso y notación a los diagramas de flujo (especialmente a los diagramas de programas), pero se adaptan a la programación orientada a objetos.

Básicamente se puede decir que el diagrama de actividades modela el flujo de actividades. Estos pueden ser procesos dentro de un sistema informático, procesos de casos de uso o procesos comerciales. Por ejemplo, la actividad “preparar una tortilla de queso” puede descomponerse en muchas actividades secundarias más pequeñas: las acciones, que pueden llevarse a cabo cronológicamente.

- **Diagrama de componentes:** Está incluido en el subgrupo de diagramas estructurales. Los diagramas de componentes UML representan las relaciones entre los

componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

En el contexto del UML, los componentes son partes modulares de un sistema independiente entre sí, que pueden reemplazarse con componentes equivalentes. Son autocontenidos y encapsulan estructuras de cualquier grado de complejidad. Los elementos encapsulados solo se comunican con los otros a través de interfaces. Los componentes no solo pueden proporcionar sus propias interfaces, sino que también pueden utilizar las interfaces de otros componentes, por ejemplo, para acceder a sus funciones y servicios. A su vez, las interfaces de un diagrama de componentes documentan las relaciones y dependencias en una arquitectura de software.

Los componentes suelen encapsular clases y, por lo tanto, también se los conoce como subformas o especializaciones de clases. Al igual que las clases, tienen una estructura compuesta y pueden definirse en más detalle por medio de atributos, métodos y operaciones, por ejemplo. Los componentes pueden ser una compilación de clases y, por ejemplo, ser implementados simultáneamente por varias clases en tiempo de ejecución. Aunque los componentes se equiparan a menudo con las clases, no son lo mismo. Si bien los componentes suelen requerir interfaces para la interacción, una clase también puede acceder directamente a un método.

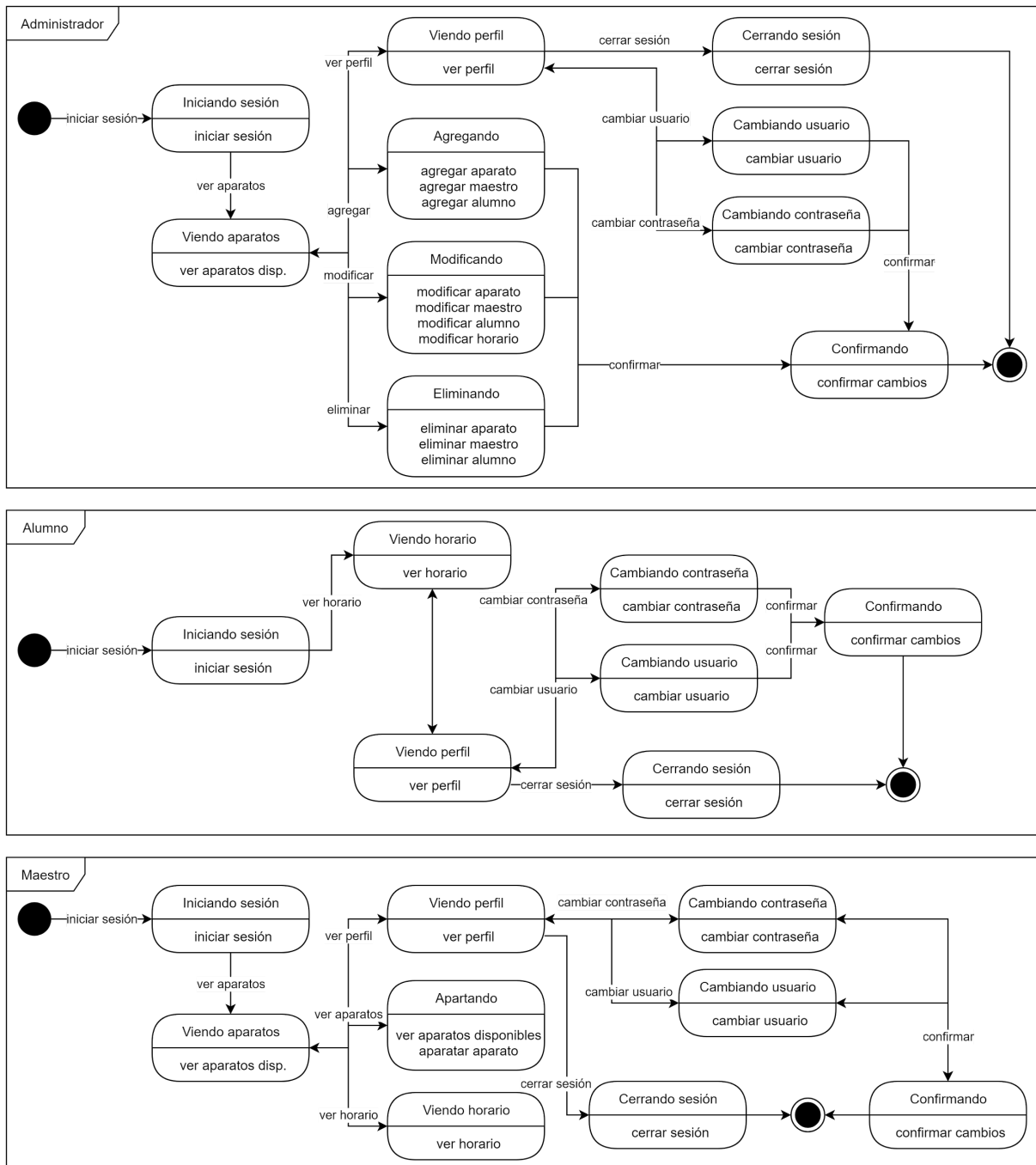
- **Diagrama de despliegue:** se utilizan para visualizar los procesadores/nodos/dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware. Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta.

Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware.

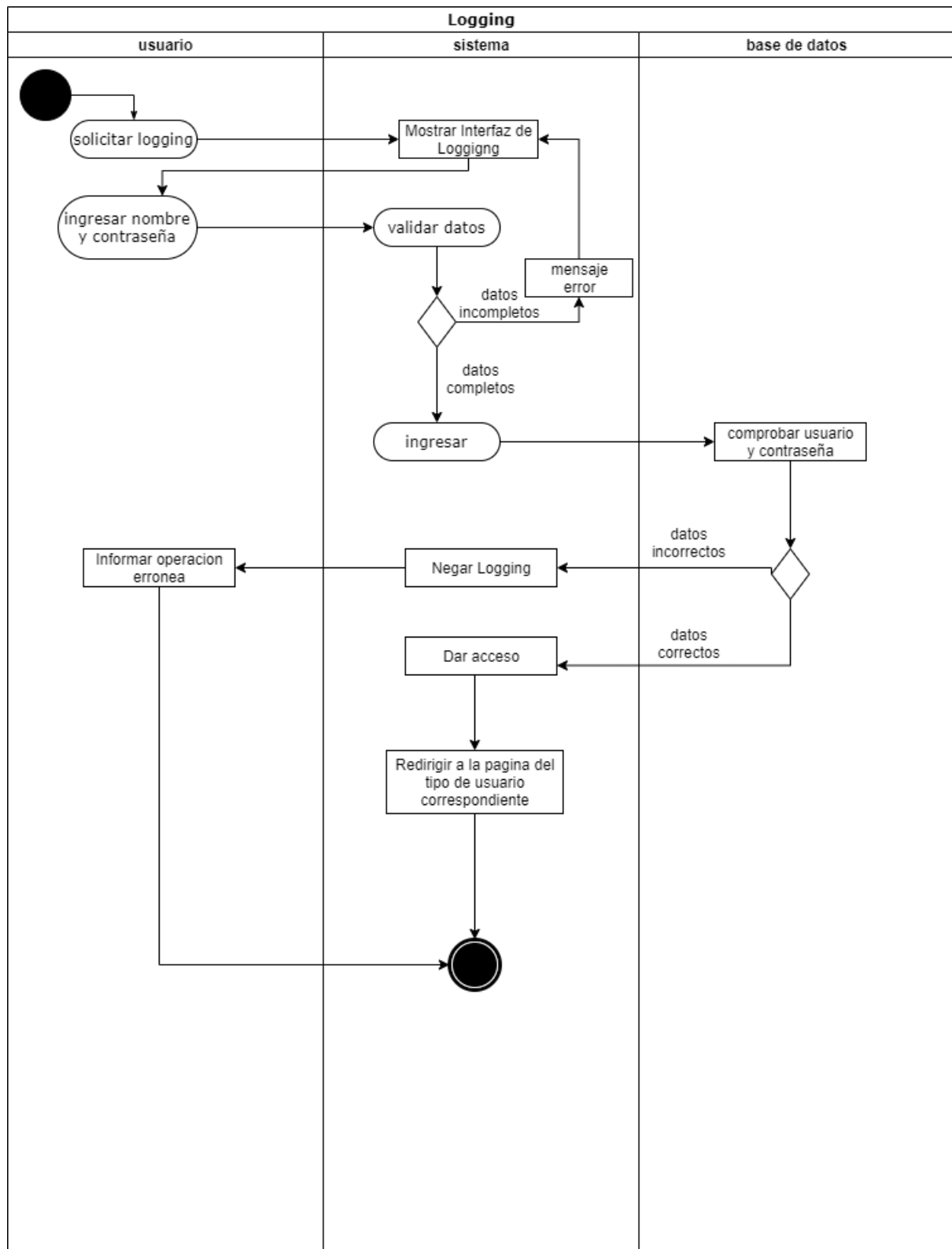
Los diagramas de despliegue ayudan a modelar la topología de hardware de un sistema en comparación con otros tipos de diagramas UML, que en su mayoría esbozan los componentes lógicos de un sistema.

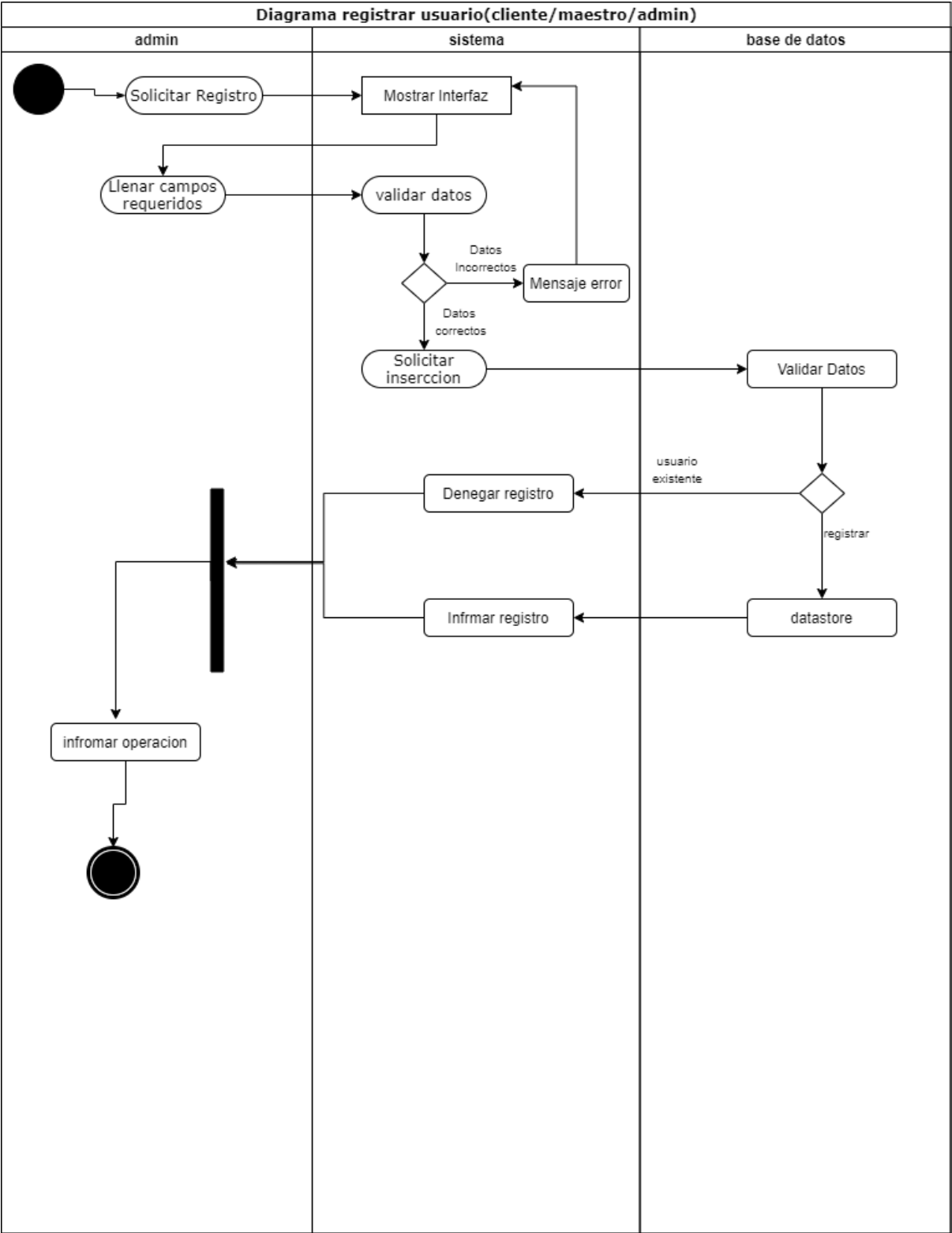
## Desarrollo

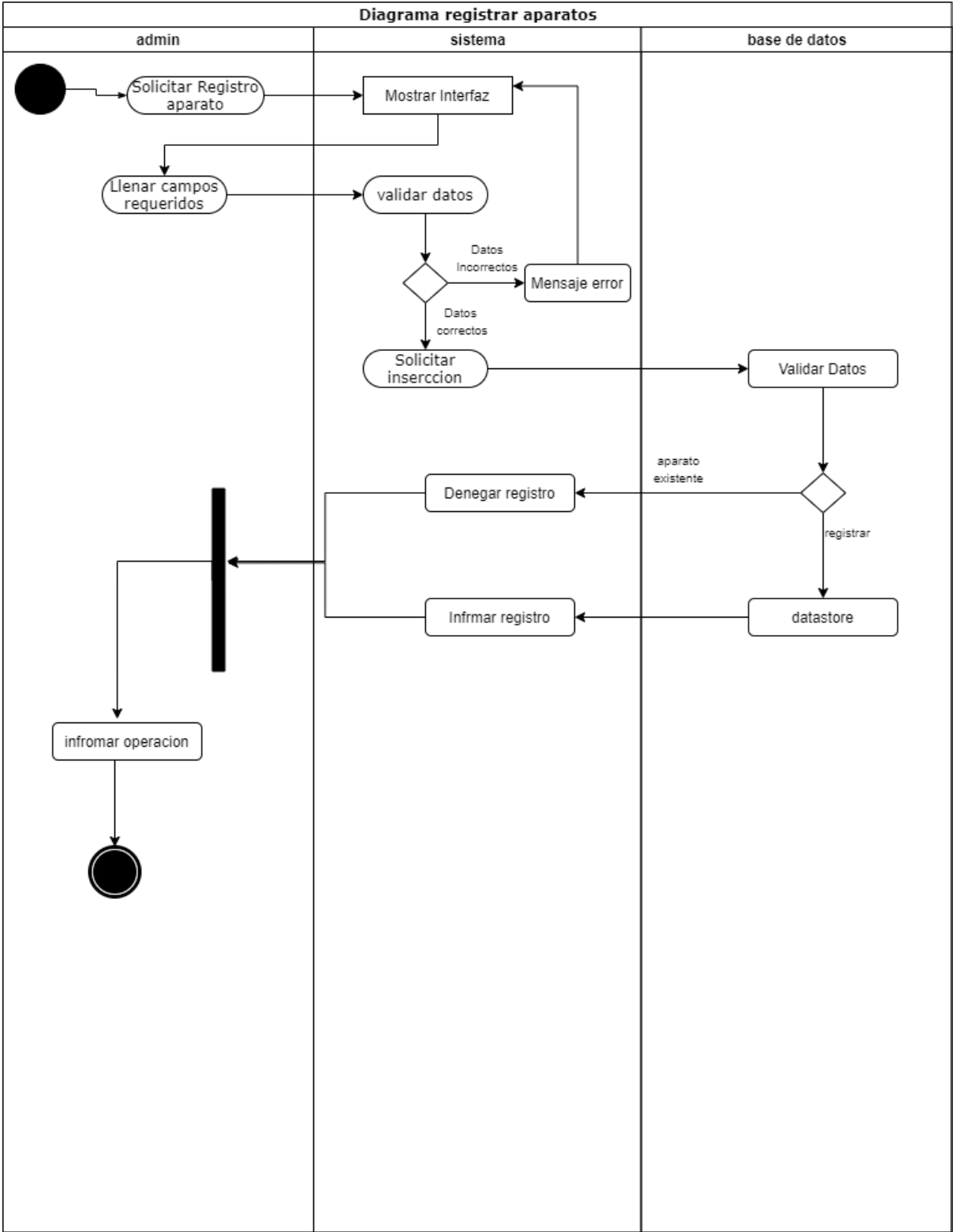
### Diagrama de estado



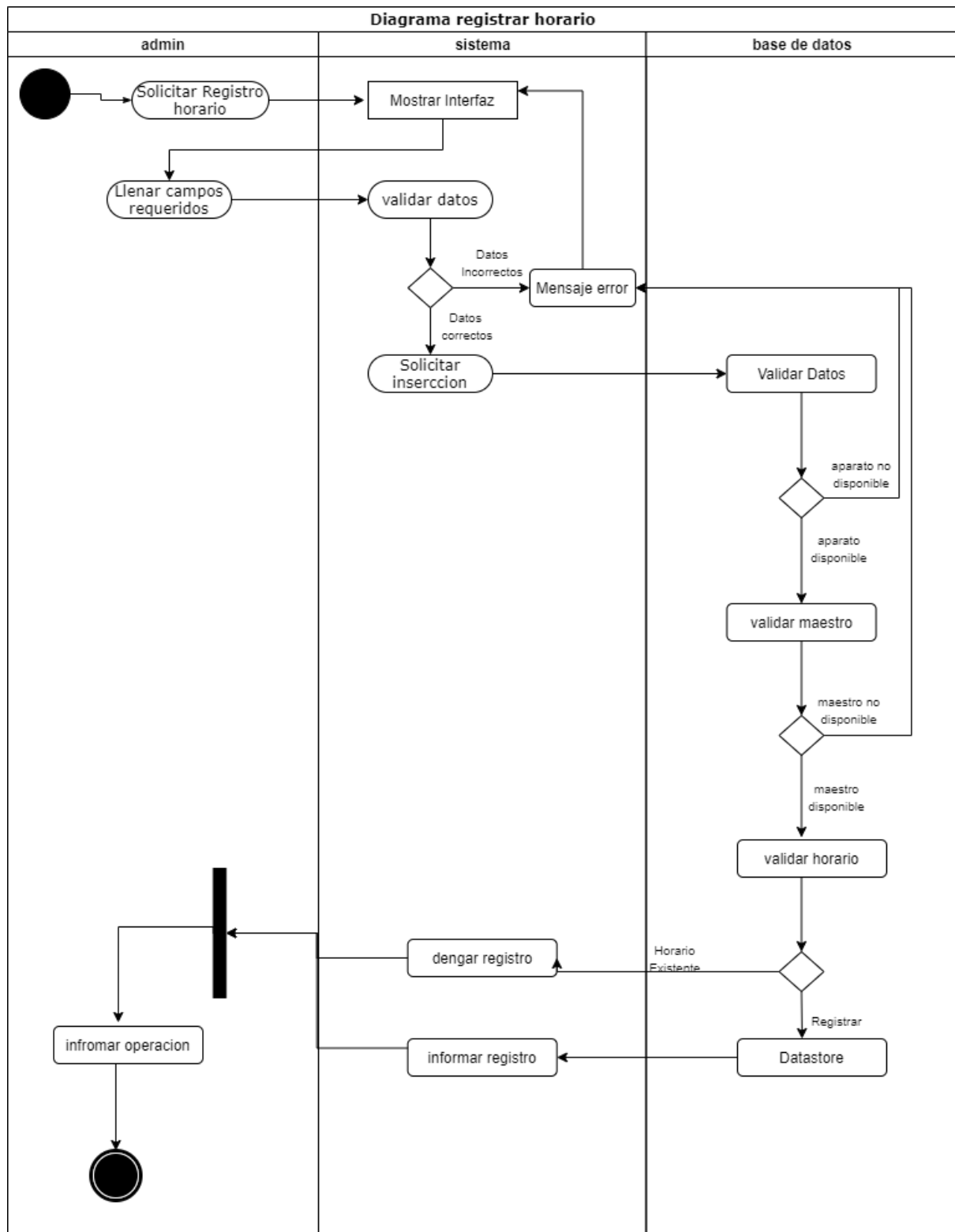
## Diagramas de actividades

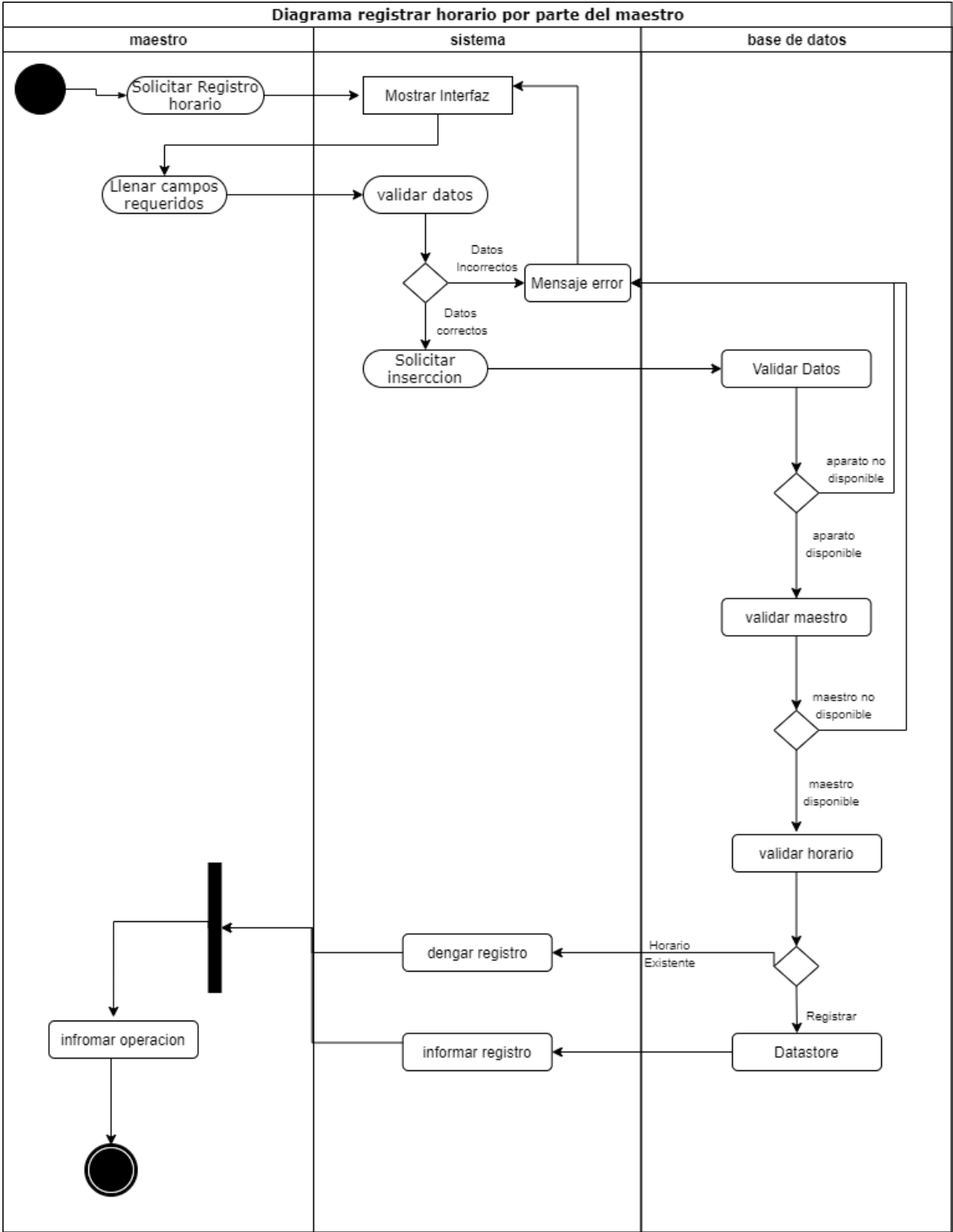


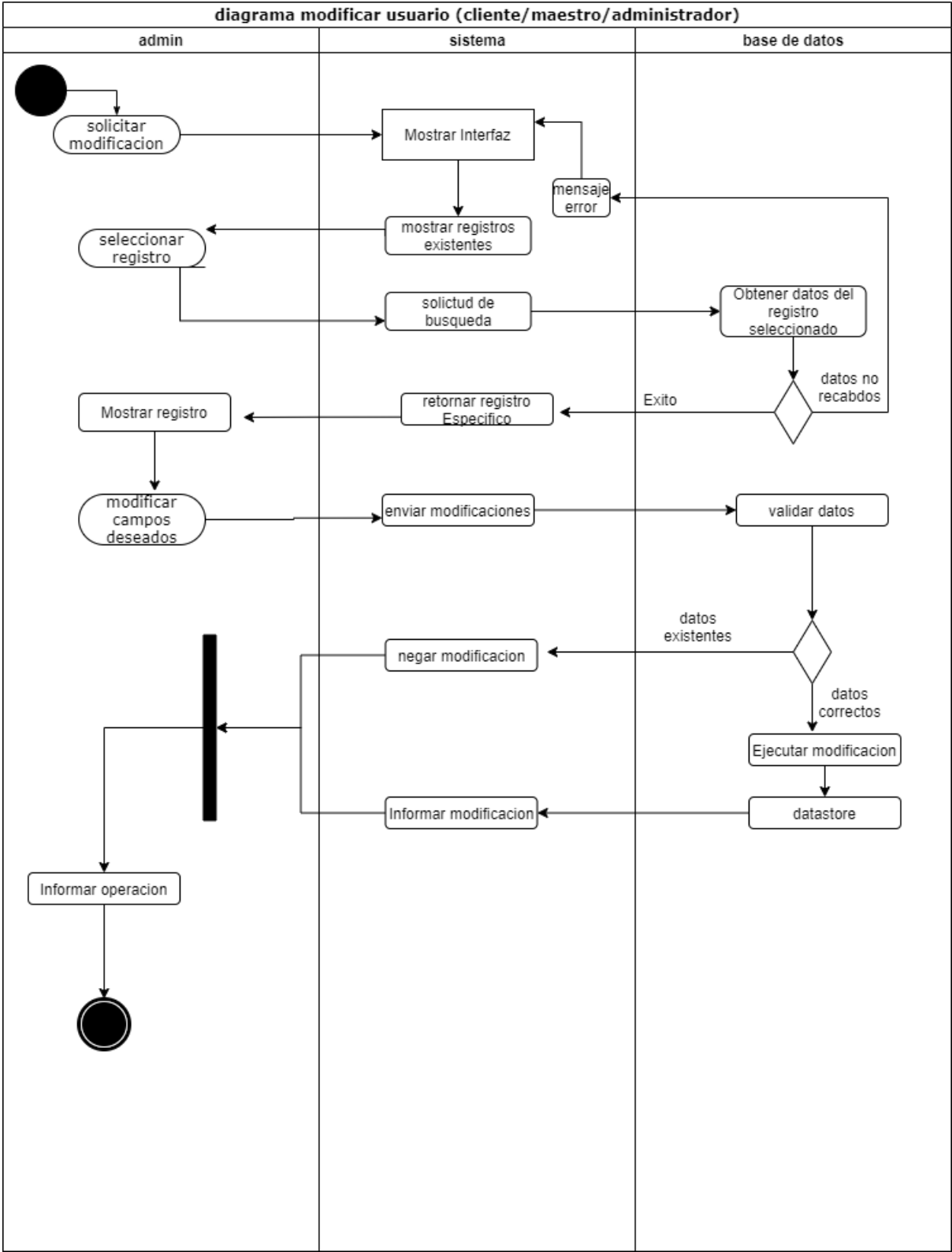


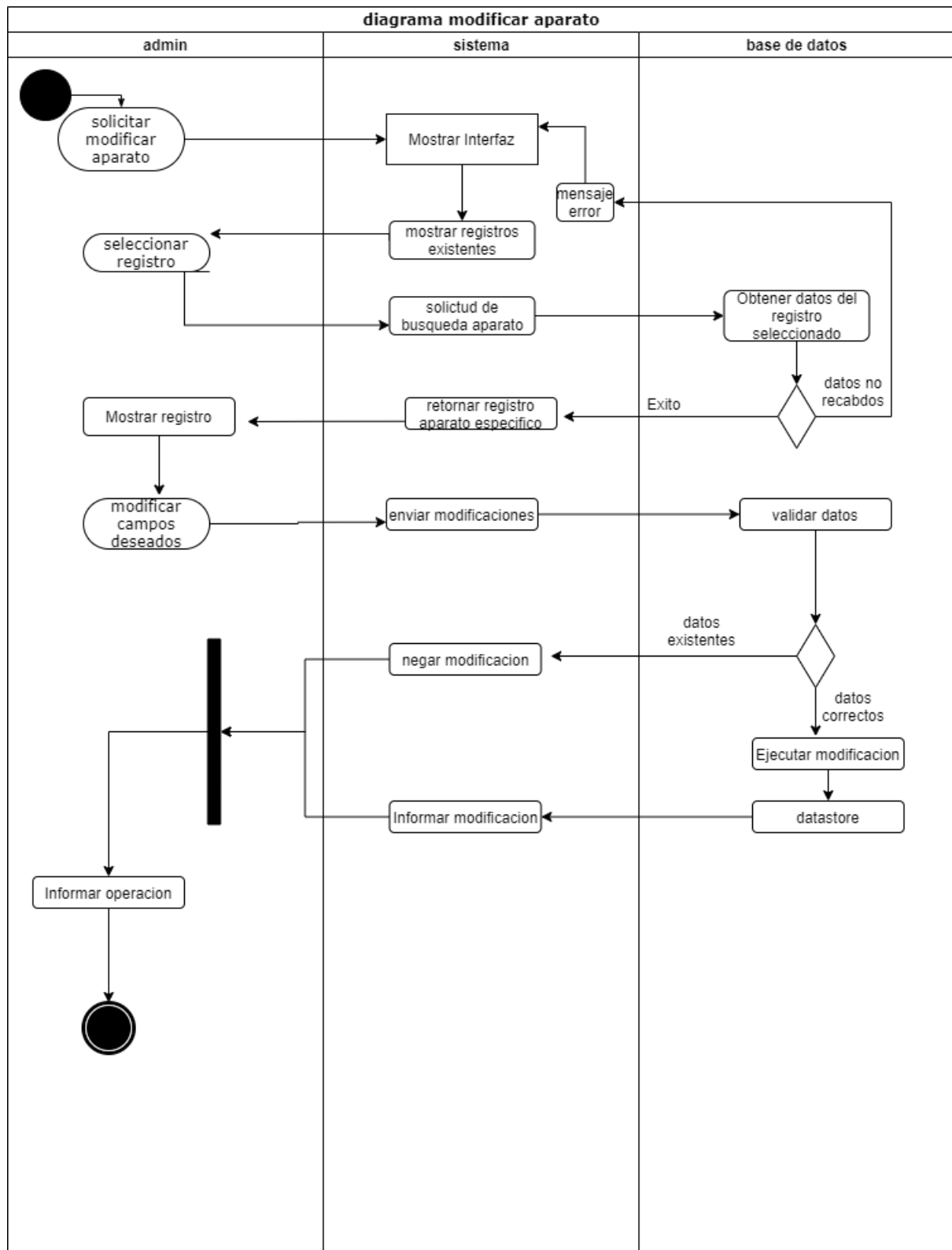








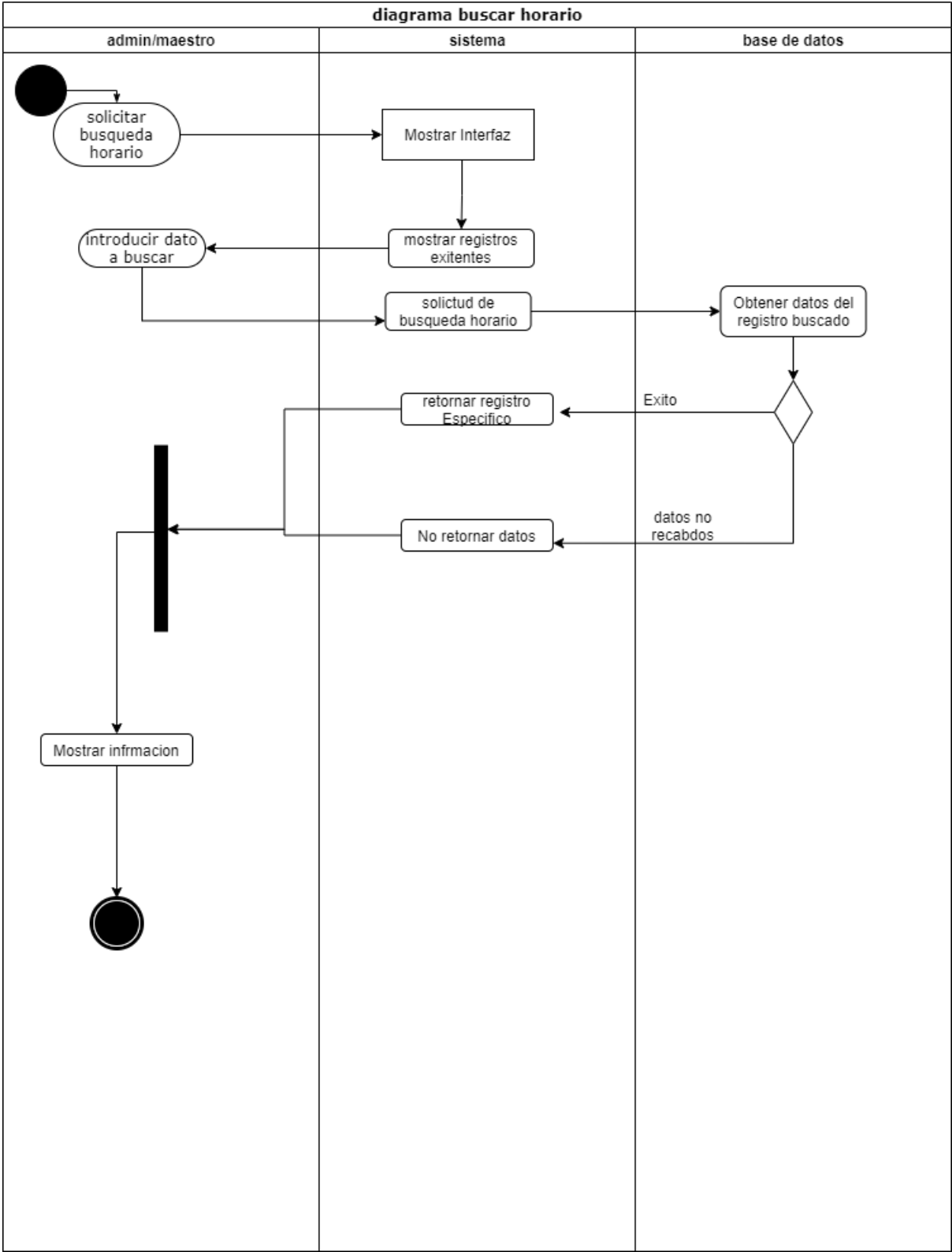




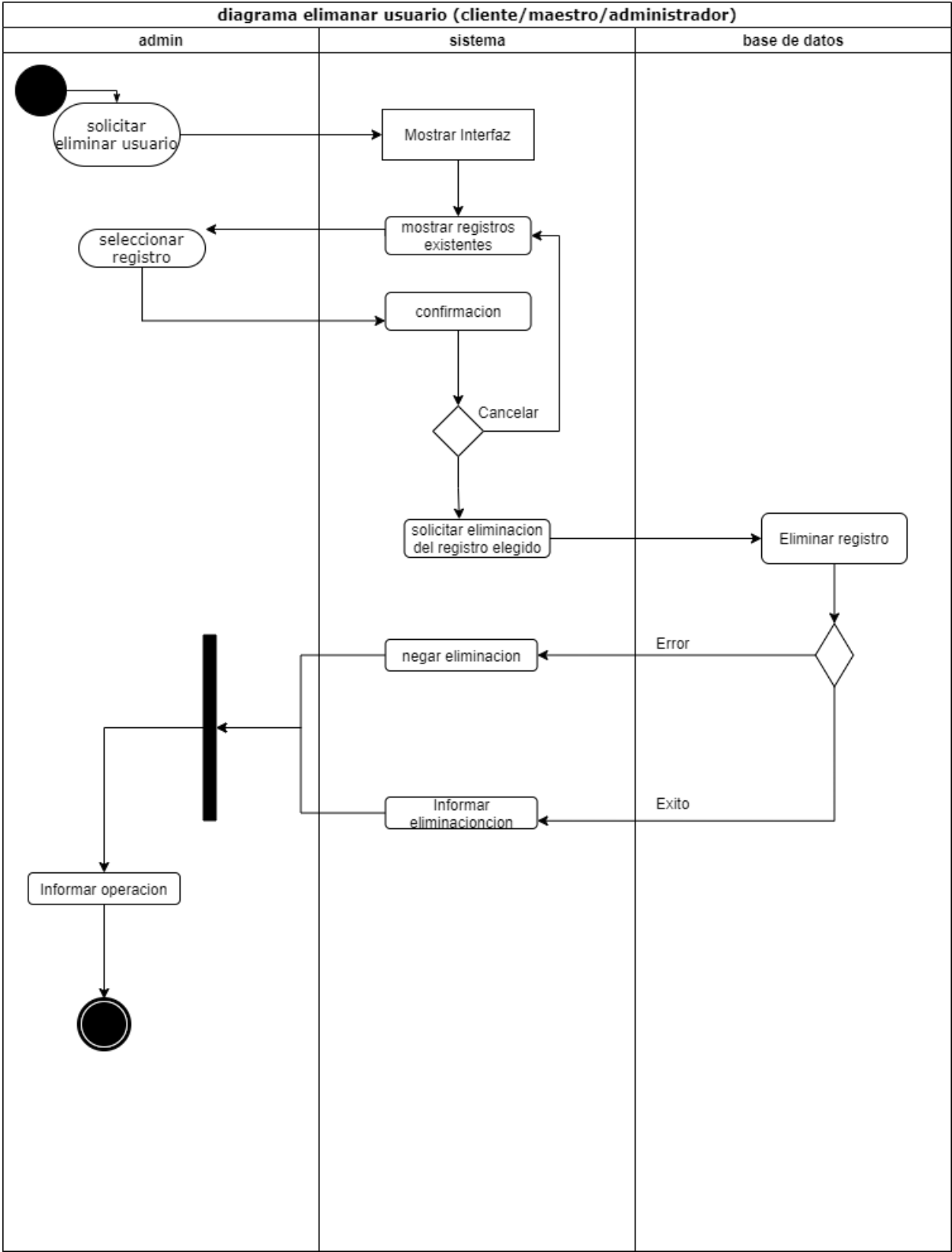


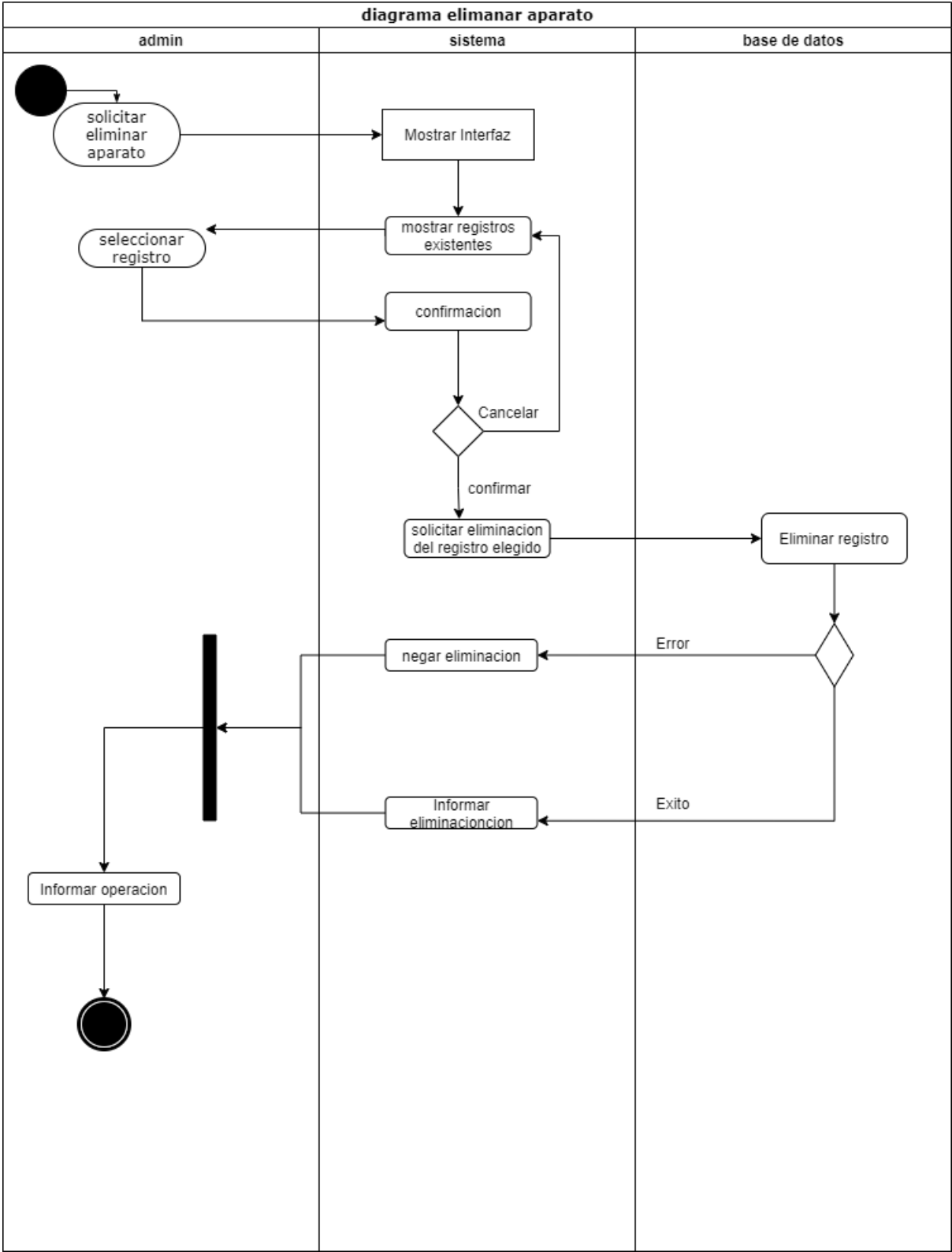


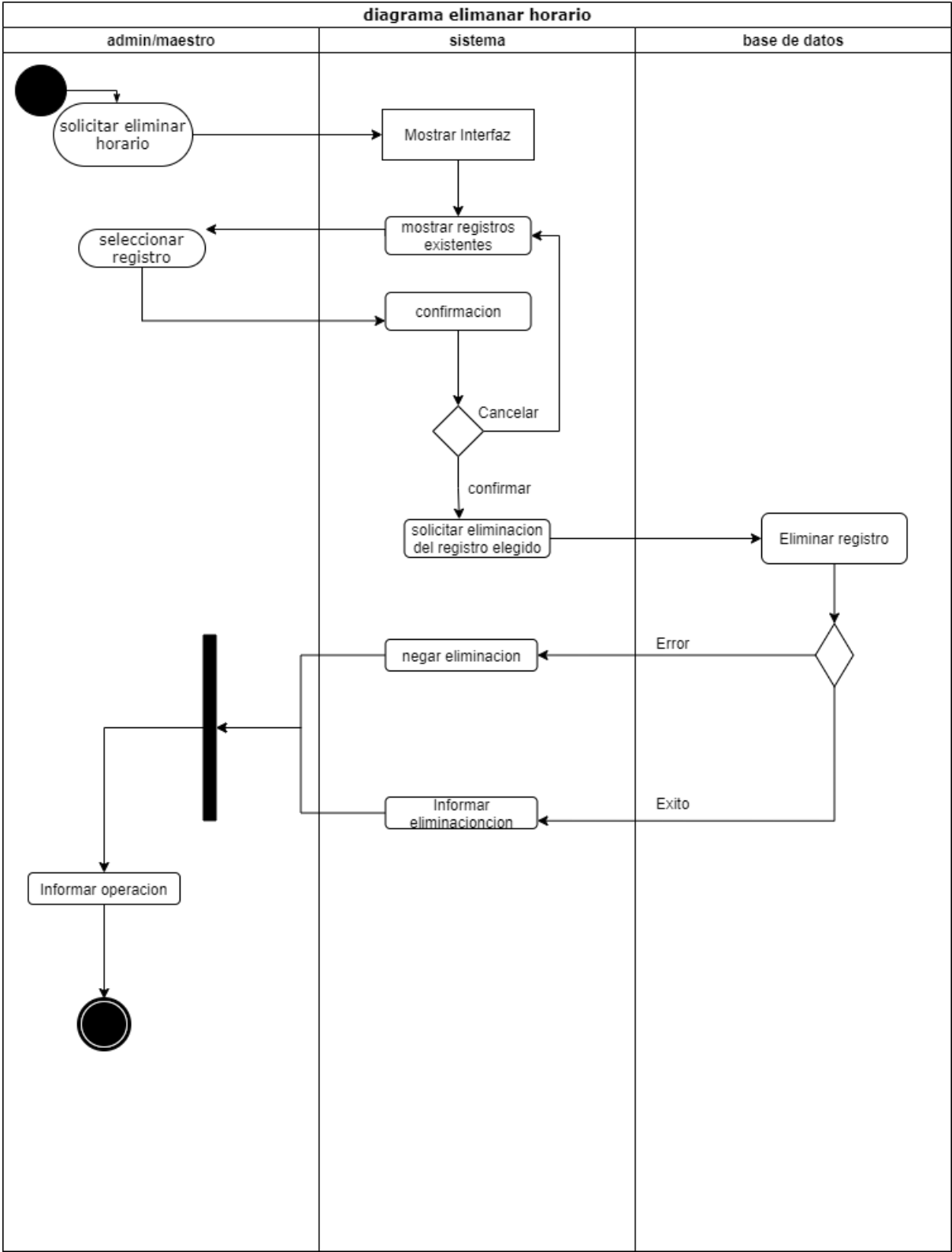




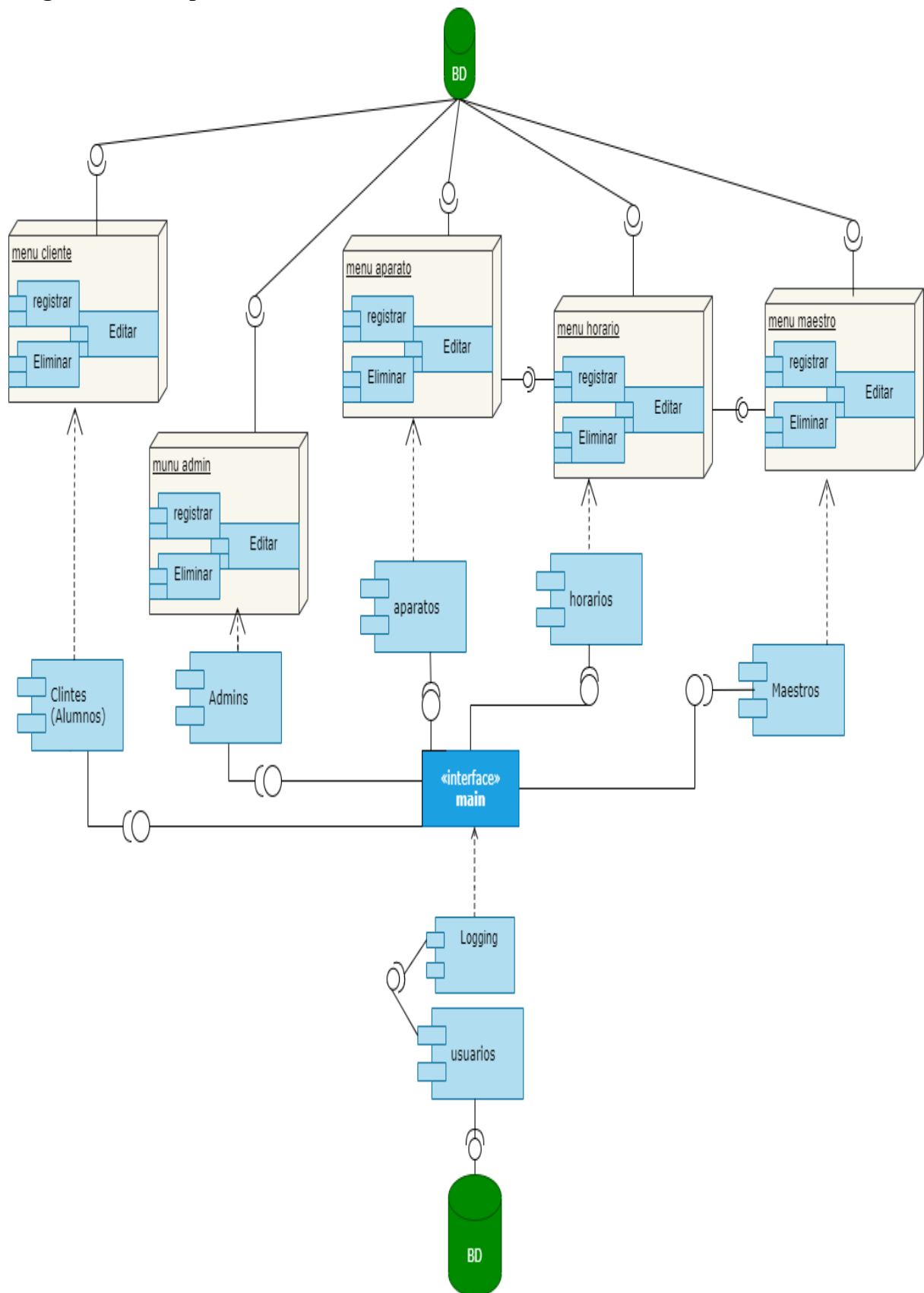






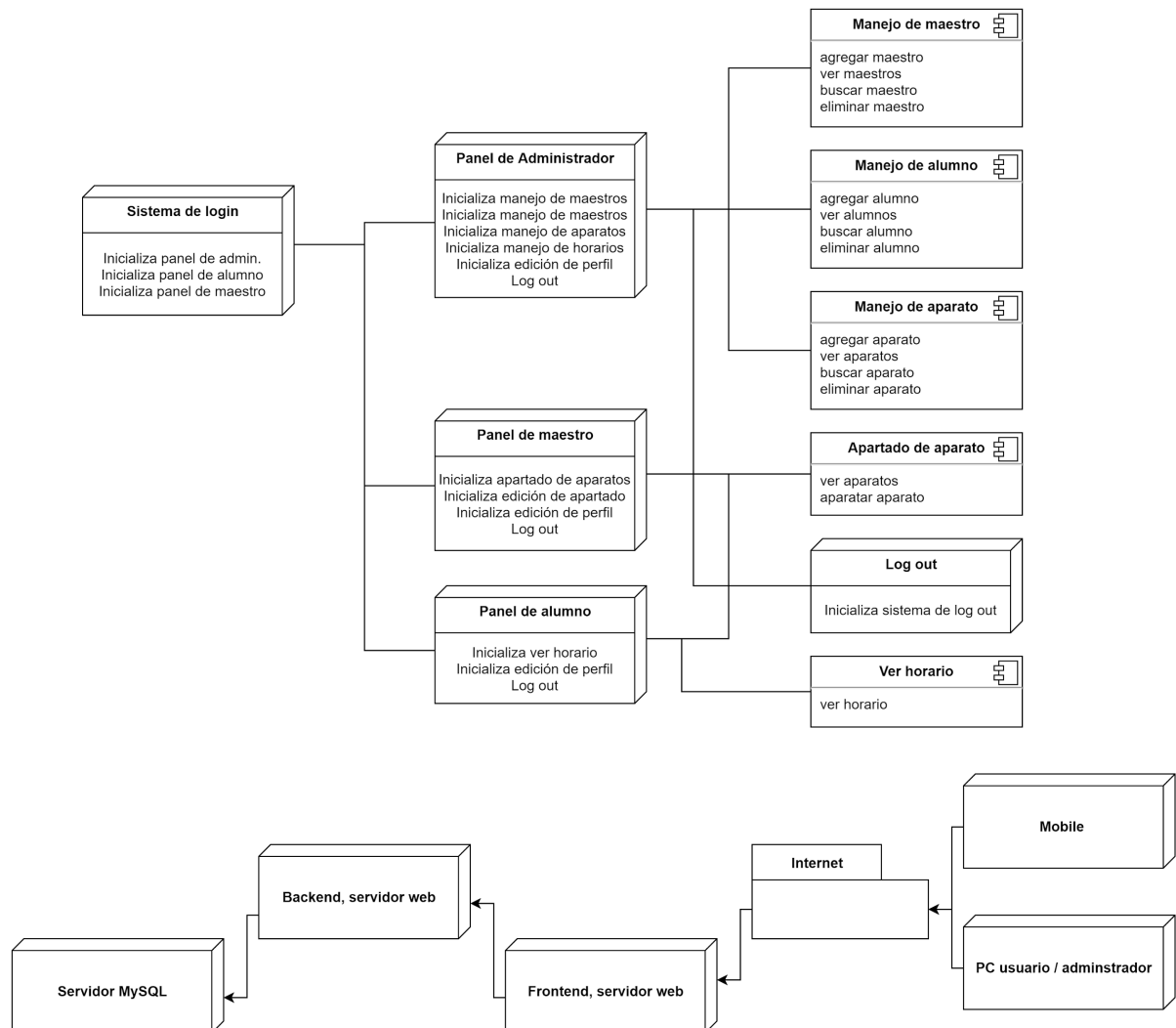


## Diagrama de componentes



## Resultados

### Diagrama de despliegue



## Conclusiones

**Daniel:** A mi parecer con estos diagramas nos damos una idea más clara y visual de las acciones y/u operaciones que un sistema tiene, además de denotar gráficamente los componentes y estructura que esto lleva y así poder también ver una sucesión de paso a paso de lo que pasa a la hora de que un usuario haga uso de una función en específico y cuáles podrían ser sus resultados.

**Emily:** Los diagramas de despliegue y de componentes nos da una visión más clara de cómo va a interactuar nuestro sistema con el hardware de la computadora y si se va a necesitar algún hardware adicional para que el sistema funcione correctamente. Mientras que los demás diagramas de estado y de actividades nos dan una mejor idea de las etapas por las que pasará nuestro sistema al realizar un proceso. Así como los demás diagramas, estos nos ayudan a comprender mejor el funcionamiento esperado del sistema.

## Referencias

- (1) IONOS (2020) “Diagrama de estado UML: Visualizar secuencias de estados de objetos” [online] IONOS. Disponible en:  
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-estado-uml/> [Obtenido el 31 de octubre 2021]
- (2) IONOS (2021) “Diagramas de actividades: El flujo de trabajo representado gráficamente” [online] IONOS. Disponible en:  
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagramas-de-actividades-uml/> [Obtenido el 31 de octubre 2021]
- (3) IONOS (2021) “Diagrama de componentes: Modelado eficiente de sistemas con módulos de software” [online] IONOS. Disponible en:  
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/> [Obtenido el 31 de octubre 2021]
- (4) Creately (2021) “La guía fácil de los diagramas de despliegue UML” [online] creately. Disponible en:  
<https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/> [Obtenido el 31 de octubre 2021]