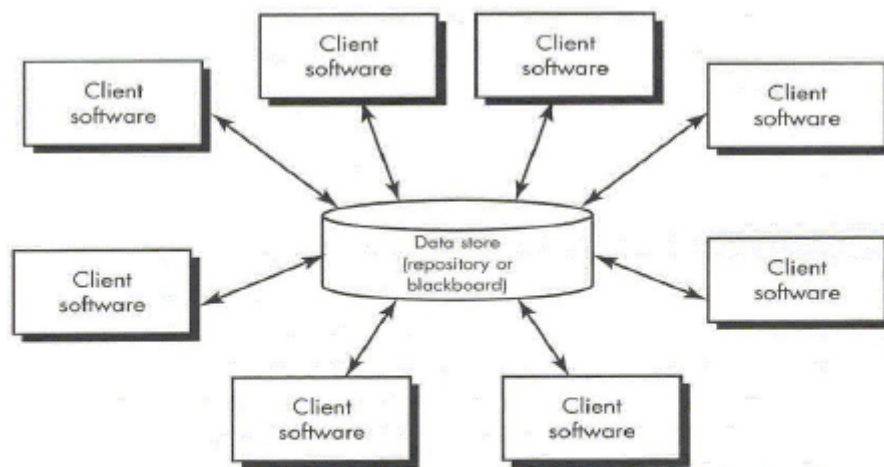


MODELO DE ARQUITECTURAS

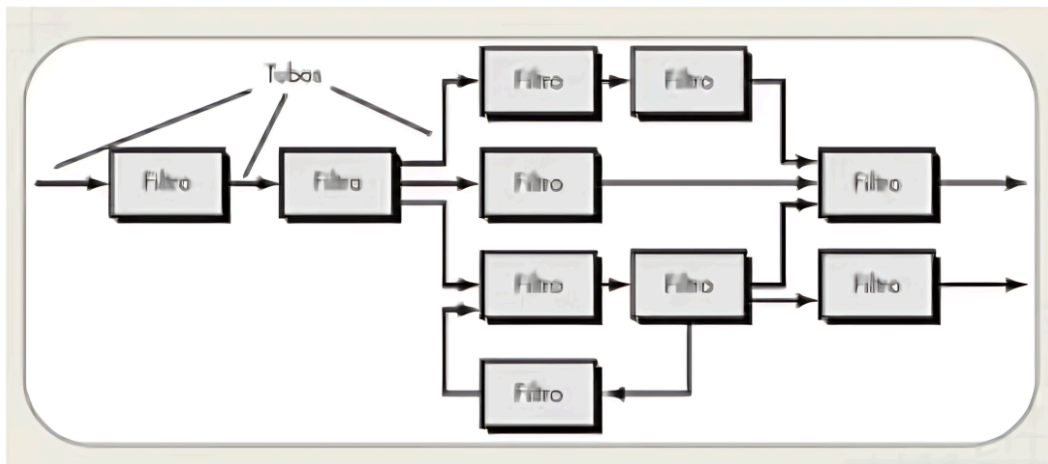
1. Centrada en datos:

Este modelo de arquitectura se enfoca en la estructura y manipulación de los datos dentro del sistema. La arquitectura está diseñada para garantizar que los datos sean almacenados, accedidos y modificados de manera eficiente y coherente. Se presta especial atención a la organización de la base de datos y a cómo los diferentes componentes del sistema interactúan con los datos.



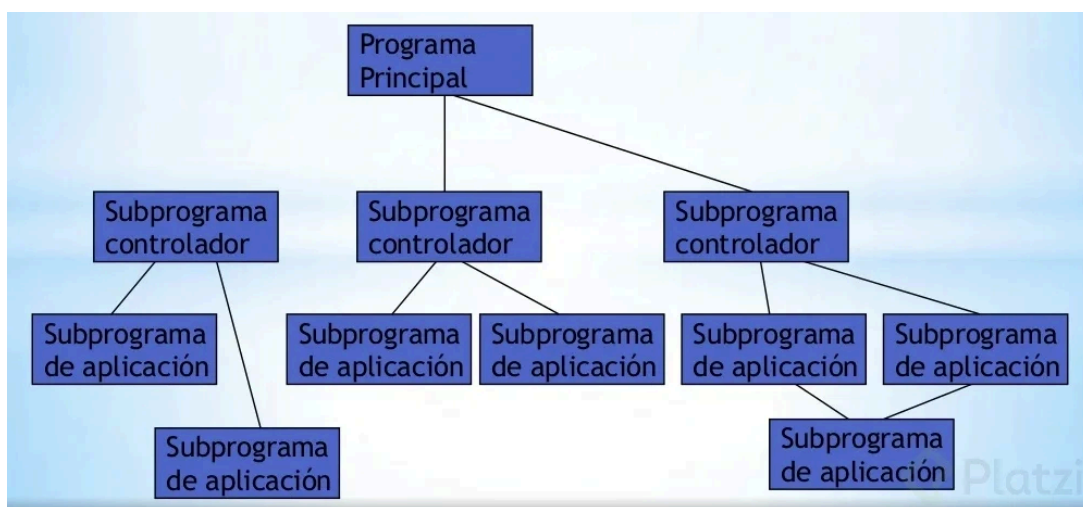
2. De flujo de datos:

En este modelo, la arquitectura se centra en cómo fluyen los datos a través del sistema y cómo se procesan en cada etapa. Se identifican los procesos que transforman los datos, así como las estructuras de datos que los almacenan temporalmente. Este enfoque es particularmente útil para sistemas donde el flujo de datos es crítico, como en sistemas de procesamiento de transacciones o sistemas de control de procesos.



3. De programa principal/subprograma:

Este modelo organiza la arquitectura en torno a un programa principal que coordina la ejecución de varios subprogramas o módulos. Cada subprograma realiza una tarea específica dentro del sistema y se comunica con el programa principal según sea necesario. Este enfoque es común en sistemas basados en procedimientos, donde la funcionalidad se divide en pequeñas unidades de trabajo independientes.



4. Orientada a objetos:

En la arquitectura orientada a objetos, el sistema se modela como una colección de objetos interconectados, cada uno con su propio estado y comportamiento. Los objetos interactúan entre sí a través de mensajes, encapsulando datos y funcionalidad relacionada. Este enfoque promueve la reutilización, la modularidad y la flexibilidad del diseño del sistema, ya que los objetos pueden ser fácilmente adaptados y extendidos para satisfacer nuevos requisitos.

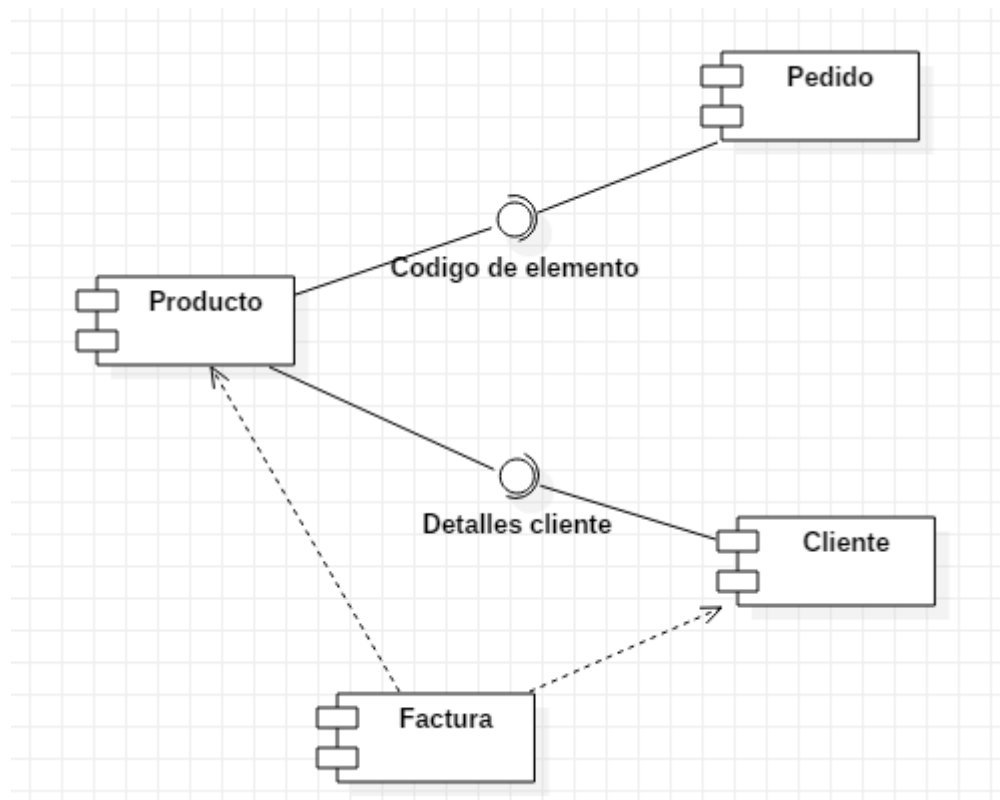


MODELO DE COMPONENTES

Modelo de Componentes

1. Visión orientada a objetos:

En este enfoque, el modelo de componentes se basa en los principios de la programación orientada a objetos (POO). Los componentes del sistema se modelan como objetos que encapsulan datos y comportamiento relacionado. Se enfatiza la reutilización de componentes a través de la herencia y la composición, lo que facilita la creación de sistemas flexibles y modulares.



2. Visión tradicional:

En contraste con la visión orientada a objetos, la visión tradicional del modelo de componentes se basa en un enfoque más estructurado. Los componentes del sistema se dividen en módulos o funciones claramente definidas. Se centra en la modularidad y la cohesión de los componentes para facilitar el mantenimiento y la evolución del sistema a lo largo del tiempo.

Modelo de Interfaz

Reglas:

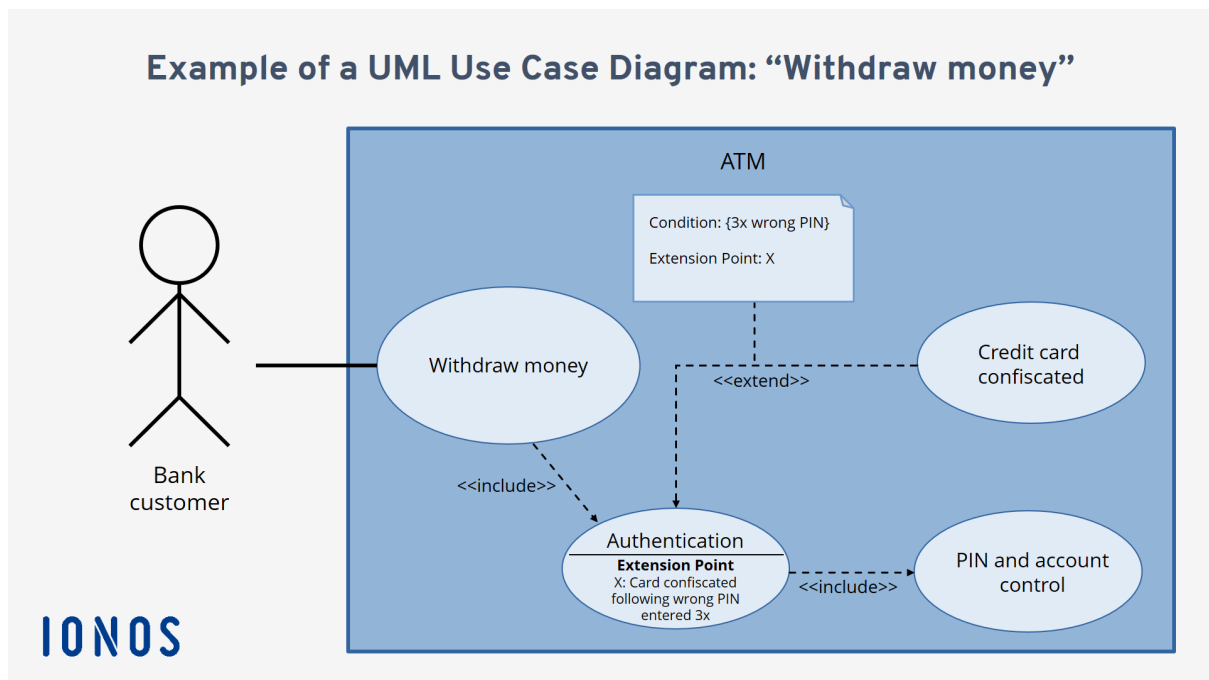
En el diseño de la interfaz de usuario, se siguen ciertas reglas para garantizar una experiencia de usuario efectiva y satisfactoria:

- Dejar el control al usuario: La interfaz debe permitir que el usuario tenga control sobre las acciones que realiza y cómo interactúa con el sistema. Esto implica proporcionar opciones claras y visibles para que el usuario pueda tomar decisiones informadas.
- Reducir la carga de memoria del usuario: La interfaz debe presentar la información de manera clara y concisa, evitando abrumar al usuario con demasiada información a la vez. Esto puede lograrse mediante la organización jerárquica de la información y el uso de técnicas como la agrupación y el filtrado de contenido.
- Hacer que la interfaz sea consistente: La interfaz debe seguir un conjunto coherente de convenciones de diseño en toda la aplicación para que el usuario pueda predecir el comportamiento del sistema y encontrar la información de manera intuitiva. Esto incluye la consistencia en la disposición de los elementos, la navegación y el uso de iconos y colores.

DIAGRAMAS DE DISEÑO. EL ESTÁNDAR UML

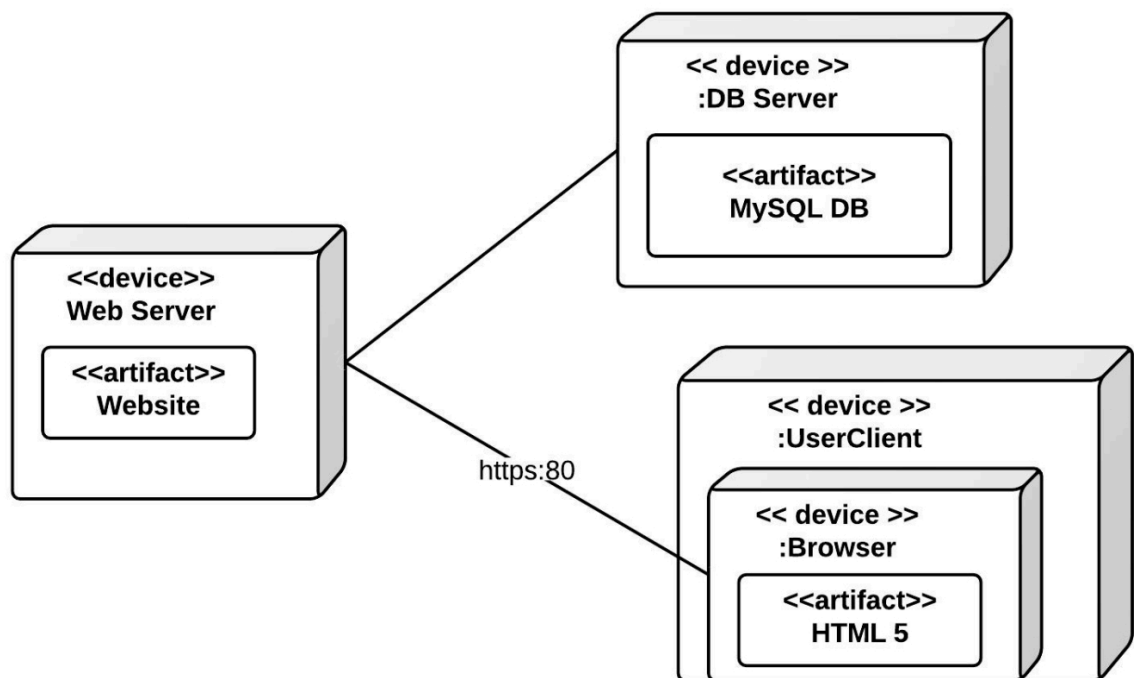
1. Diagramas de Casos de Uso:

Los diagramas de casos de uso modelan las interacciones entre actores externos y el sistema, identificando los casos de uso y sus relaciones. Los actores representan usuarios o sistemas externos que interactúan con el sistema, mientras que los casos de uso representan las funcionalidades específicas proporcionadas por el sistema. Las relaciones entre actores y casos de uso ilustran quién puede realizar qué acciones.



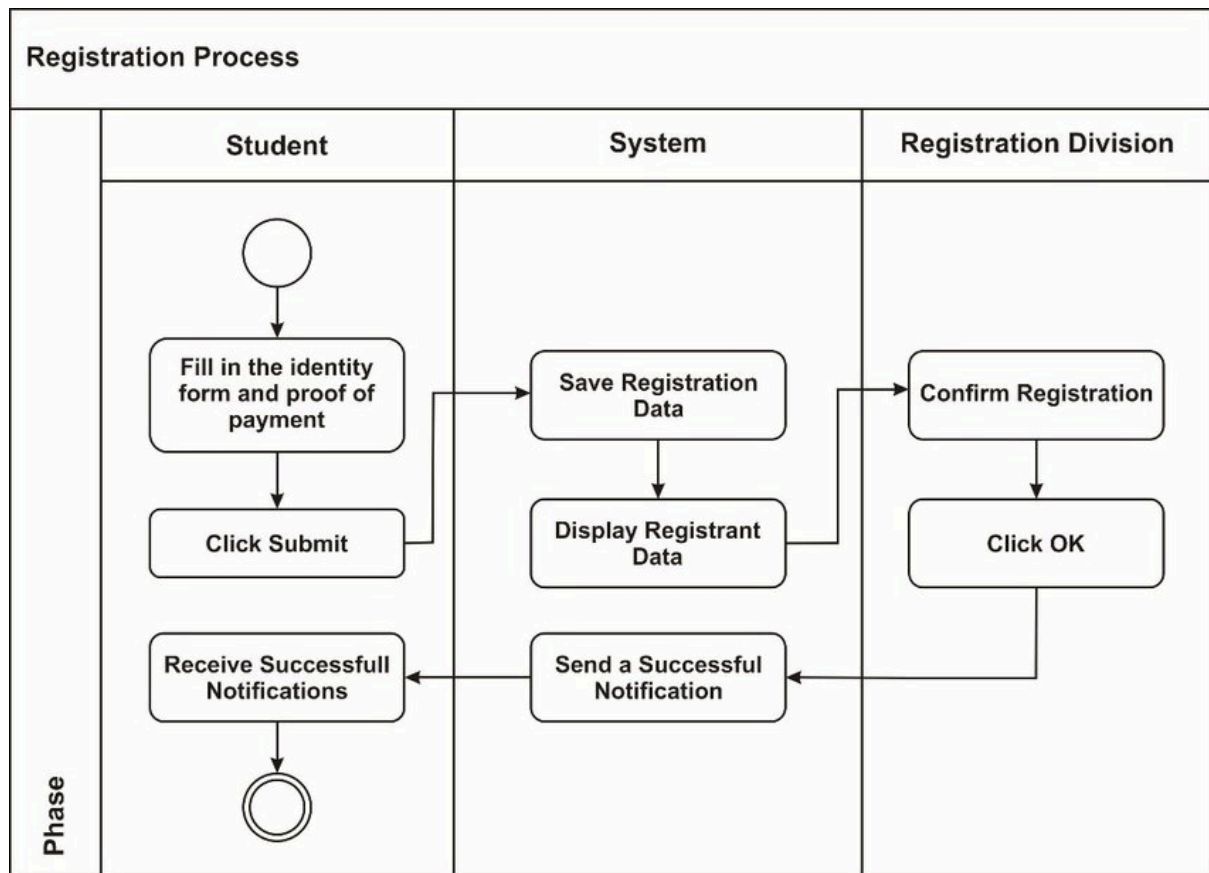
2. Diagramas de Implementación:

Los diagramas de implementación ilustran cómo se distribuyen los artefactos de software en nodos de hardware dentro de un sistema. Los nodos representan dispositivos de hardware o entornos de ejecución de software, y los artefactos representan componentes de software o archivos. Las relaciones de implementación muestran cómo se despliegan los artefactos en los nodos.



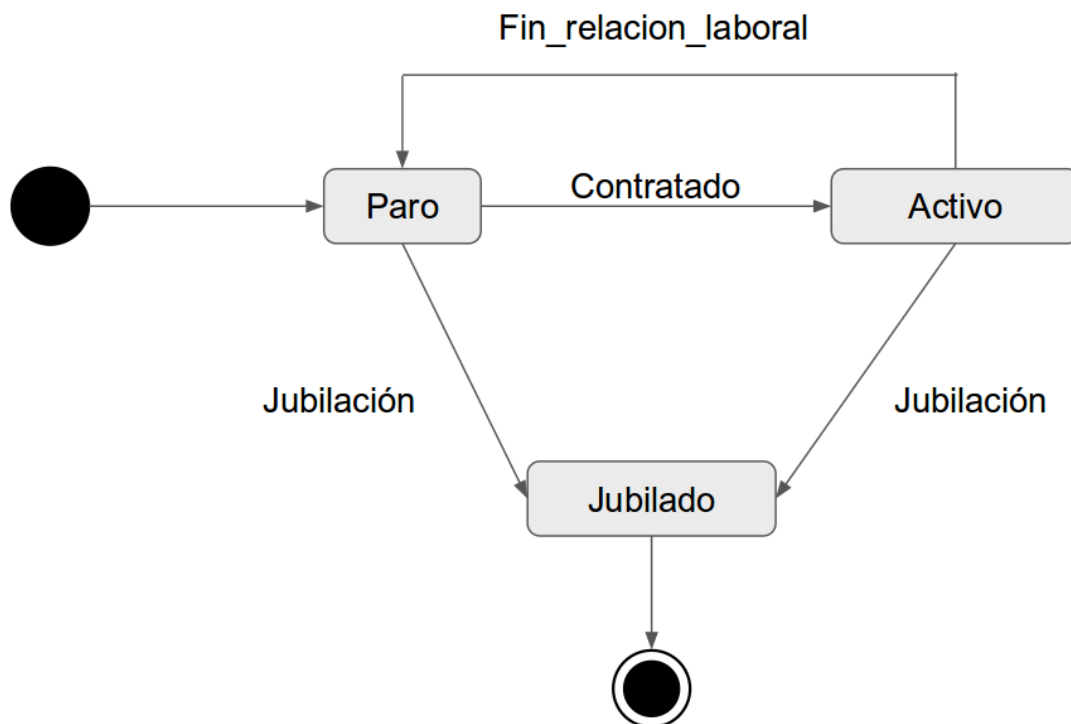
3. Diagramas de Actividad:

Los diagramas de actividad describen el flujo de control entre diferentes actividades o estados dentro del sistema. Las actividades representan acciones o pasos realizados por el sistema o sus usuarios, mientras que las transiciones representan el flujo de control entre actividades. Puntos de decisión, bifurcaciones y uniones se utilizan para modelar el comportamiento condicional y flujos paralelos.



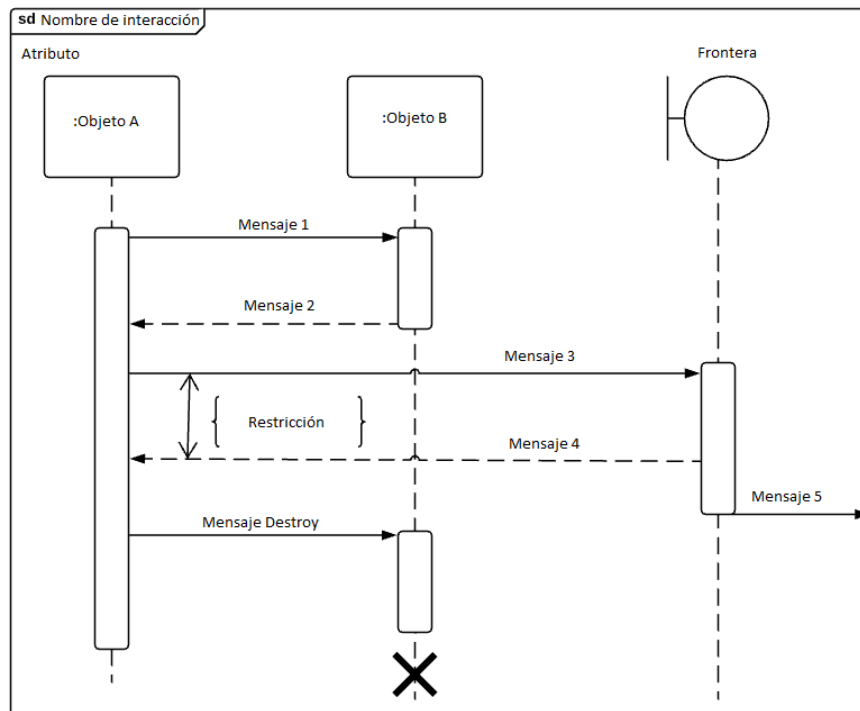
4. Diagramas de Estado:

Los diagramas de estado representan los diferentes estados en los que puede encontrarse un objeto o sistema, así como las transiciones entre estos estados. Los estados representan condiciones o situaciones durante la vida útil de un objeto, mientras que las transiciones representan los eventos o disparadores que provocan el cambio de estado. Los diagramas de estado son especialmente útiles para modelar el comportamiento de sistemas complejos con múltiples estados y transiciones.



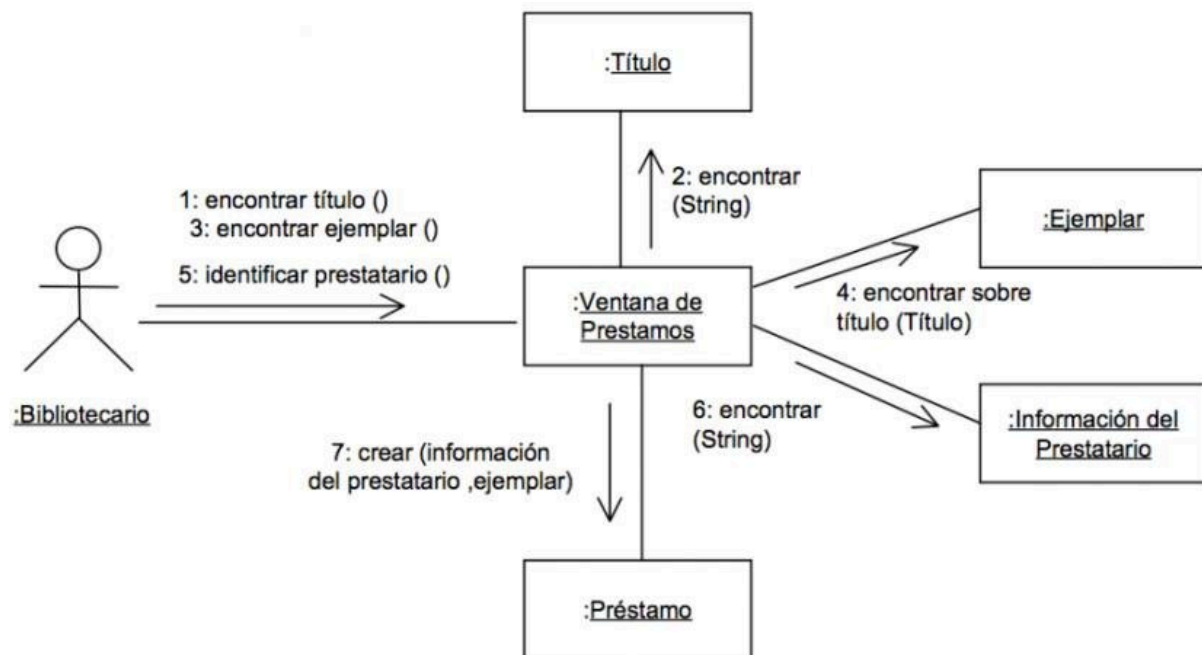
5. Diagramas de Secuencia:

Los diagramas de secuencia ilustran la interacción entre objetos en un sistema a lo largo del tiempo, mostrando el orden en el que se intercambian mensajes entre objetos. Los objetos representan instancias de clases o componentes en el sistema, mientras que los mensajes representan la comunicación entre objetos. Las líneas de vida muestran la duración de los objetos, y las activaciones representan cuándo los objetos están procesando activamente mensajes.



6. Diagramas de Comunicación:

Los diagramas de comunicación, también conocidos como diagramas de colaboración, son similares a los diagramas de secuencia pero se centran en la organización estructural de los objetos y sus asociaciones. Los objetos representan instancias de clases o componentes.



Documentación

1. Introducción:

La introducción proporciona una visión general del documento de documentación, presentando su propósito y alcance. También puede incluir información sobre el sistema en cuestión, su contexto y su importancia.

2. Descripción general del sistema:

Esta sección describe las características y funcionalidades principales del sistema en un nivel alto. Proporciona una visión general de lo que hace el sistema, quiénes son sus usuarios principales y cómo se espera que beneficie a sus usuarios o a la organización en general.

3. Arquitectura del sistema:

En esta parte se detalla la estructura y organización del sistema desde un punto de vista arquitectónico. Esto incluye información sobre los diferentes componentes del sistema, cómo están interconectados y cómo se comunican entre sí. También puede abordar temas como la escalabilidad, la seguridad y el rendimiento del sistema.

4. Diseño de datos:

Aquí se describe la estructura y organización de los datos dentro del sistema. Esto puede incluir información sobre las bases de datos utilizadas, los modelos de datos empleados, los esquemas de bases de datos, así como las relaciones y restricciones entre los diferentes conjuntos de datos.

5. Diseño de componentes:

Esta sección detalla la implementación y organización de los componentes del sistema. Se describe cómo se dividen las funcionalidades del sistema en componentes más pequeños y cómo estos componentes interactúan entre sí para lograr los objetivos del sistema. También puede abordar temas como la modularidad, la cohesión y el acoplamiento de los componentes.

6. Diseño de interfaz:

Aquí se describe el diseño visual y funcional de la interfaz de usuario del sistema. Se abordan aspectos como la disposición de los elementos de la interfaz, el flujo de navegación, la interacción del usuario y la consistencia visual. También puede incluir detalles sobre los estándares de diseño utilizados y las pautas de accesibilidad para garantizar una experiencia de usuario óptima.