Diagramas de estructura

Diagrama de clases

Es un diagrama de estructura estática que muestra las clases del sistema, sus atributos, operaciones (o métodos) y las relaciones entre objetos. Es como un plano del diseño del sistema.

- **Clase**: representa las estructuras de la aplicación. Cada clase está dividida en cuatro compartimentos:
    - El nombre de la clase indica lo que es la clase y no lo que hace.
    - Los *atributos* de la clase determinan las características de la clase.
    - Las *operaciones* de la clase representan las acciones posibles en la clase.

Diagrama de despliegue

Se utilizan normalmente para visualizar el hardware y el software físico de un sistema. usandolo puedes entender cómo el sistema des desplegara físicamente en el hardware.

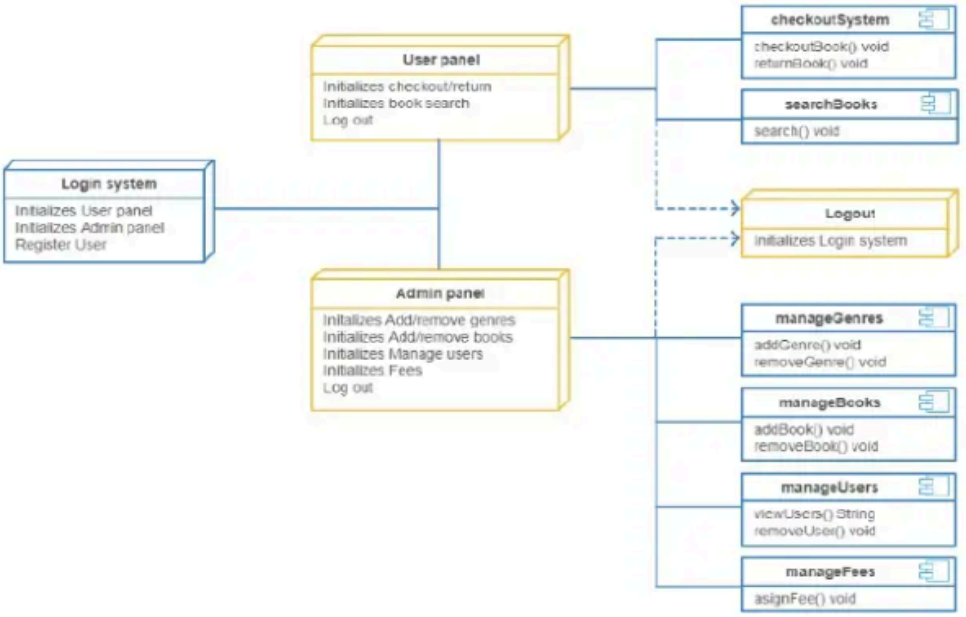


Ilustración 1. Diagrama de despliegue para el sistema de Gestión de Bibliotecas

Diagrama de objetos

Se enfoca en los atributos de un conjunto de objetos y cómo esos objetos se relacionan entre sí. Por ejemplo, en el siguiente diagrama de objetos, las tres cuentas bancarias están ligadas al mismo banco. Los títulos de clase muestran el tipo de cuentas (ahorros, corriente y tarjeta de crédito) que un cliente dado podría tener con este banco en particular. Los atributos de clase son diferentes para cada tipo de cuenta. Por ejemplo, el objeto de tarjeta de crédito tiene un límite de crédito, mientras que las cuentas de ahorros y corriente tienen tasas de interés.

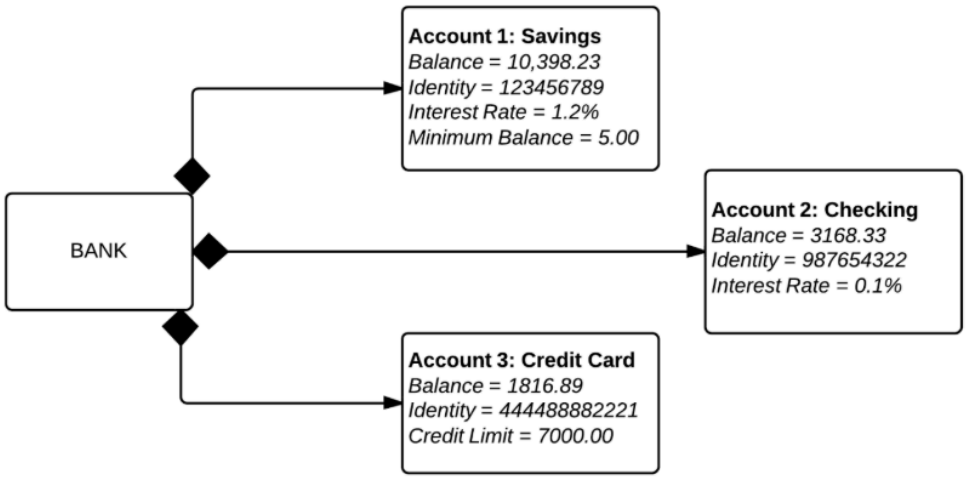


Diagrama de paquetes

Los diagramas de paquetes son diagramas estructurales que se emplean para mostrar la organización y disposición de diversos elementos de un modelo en forma de paquetes. Un paquete es una agrupación de elementos UML relacionados, como diagramas, documentos, clases o, incluso, otros paquetes. Cada elemento está anidado dentro de un paquete, que se representa como una carpeta de archivos dentro del diagrama, y que luego se organiza jerárquicamente dentro del diagrama.

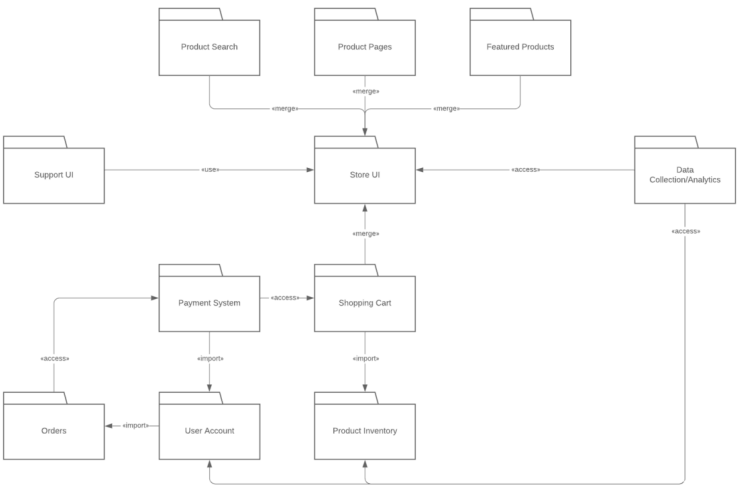


Diagrama de estructura

Brinda una vista general lógica de todo o parte de un sistema de software. Actúa como una mirada al interior de un clasificador estructurado determinado a fin de definir sus clases de configuración, interfaces, paquetes y las relaciones entre ellos a un micronivel. Permite que los

usuarios vean exactamente qué contiene un objeto a fin de especificar cómo encajan las distintas propiedades para producir un cierto comportamiento.

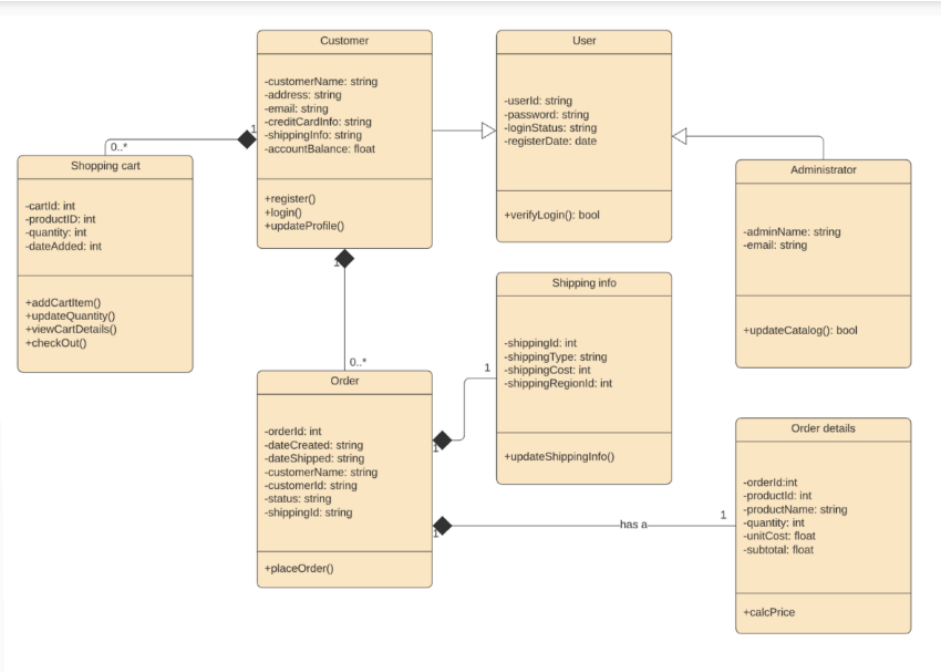


Diagrama de componentes

El término "componente" se refiere a un módulo de clases que representa sistemas o subsistemas independientes con la capacidad de interactuar con el resto del sistema. Ofrece una visión general de su sistema de software. Comprender el comportamiento exacto de cada componente de su software le ayudará a ser un mejor desarrollador. Los diagramas de componentes pueden describir sistemas de software implementados en cualquier lenguaje o estilo de programación.

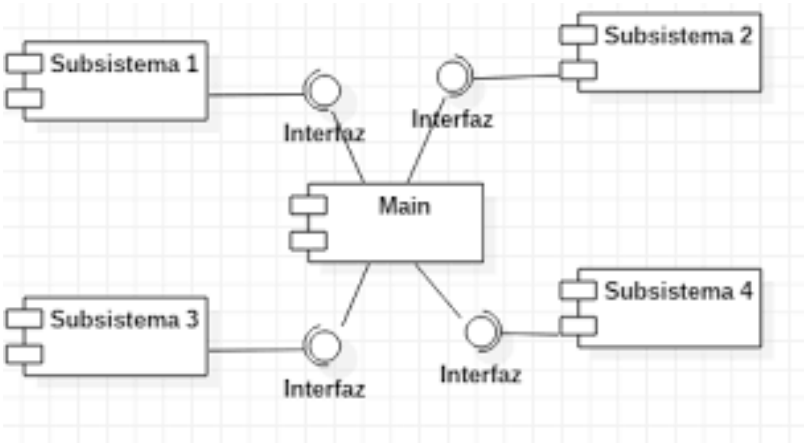


Diagrama de Actividad

Los diagramas de actividades presentan una serie de beneficios para los usuarios. Considera crear un diagrama de actividades para:

- Demostrar la lógica de un algoritmo.
- Describir los pasos realizados en un caso de uso UML.
- Ilustrar un proceso de negocios o flujo de trabajo entre los usuarios y el sistema.
- Simplificar y mejorar cualquier proceso clarificando casos de uso complicados.
- Modelar elementos de arquitectura de software, tales como método, función y operación.

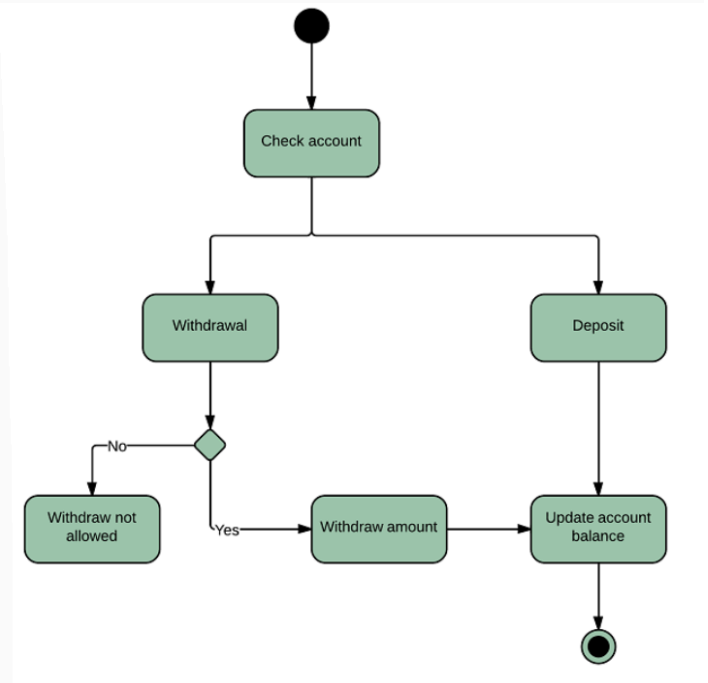


Diagrama de Máquina de estados

Una máquina de estados es cualquier dispositivo que almacena el estado de un objeto en un momento dado y puede cambiar el estado o causar otras acciones según la entrada que reciba. Estados se refiere a las diferentes combinaciones de información que un objeto puede mantener, no la forma en que el objeto se comporta. Para comprender los diferentes estados de un objeto, podrías visualizar todos los estados posibles y mostrar cómo un objeto llega a cada estado. Los diagramas de estado representan principalmente estados y transiciones. Los estados se representan con rectángulos de esquinas redondeadas que se etiquetan con el nombre del estado. Las transiciones se marcan con flechas que fluyen de un estado a otro, mostrando cómo cambian los estados.

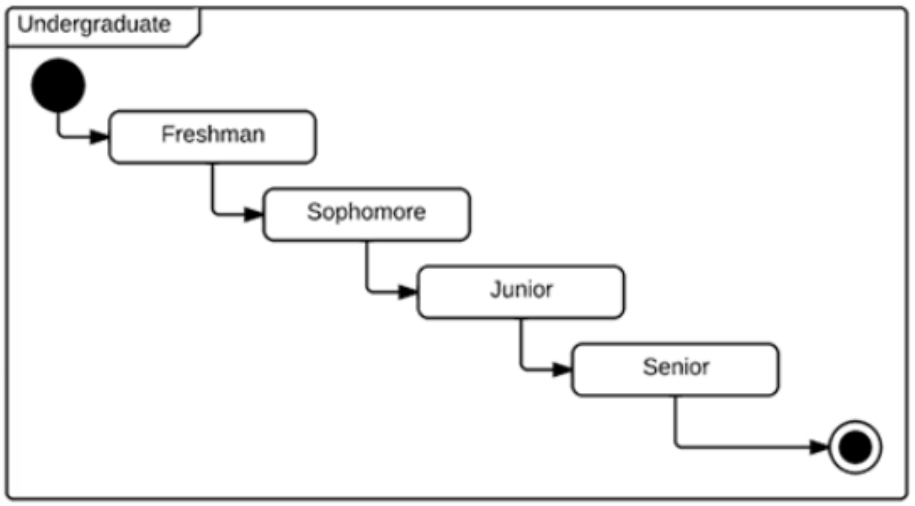


Diagrama de casos de uso

Un diagrama de casos de uso es la forma principal de los requisitos de un sistema o software para un nuevo programa de software en desarrollo. Los casos de uso especifican el comportamiento esperado (qué), y no el método exacto para lograrlo (cómo). Una vez especificados, los casos de uso pueden representarse tanto de forma textual como visual (es decir, el diagrama de casos de uso). Un concepto clave del modelado de casos de uso es que nos ayuda a diseñar un sistema desde la perspectiva del usuario final. Es una técnica eficaz para comunicar el comportamiento del sistema en los términos del usuario, especificando todo el comportamiento del sistema visible externamente.

Un diagrama de casos de uso suele ser simple. No muestra el detalle de los casos de uso:

- Solo resume algunas de las relaciones entre casos de uso, actores y sistemas.
- No muestra el orden en que se realizan los pasos para lograr los objetivos de cada caso de uso.

La relación de inclusión (<<include>>) agrega funcionalidad adicional no especificada en el caso de uso base. Se utiliza para incorporar comportamiento común de un caso de uso "incluido" en un caso de uso "base", con el fin de fomentar la reutilización de dicho comportamiento común.

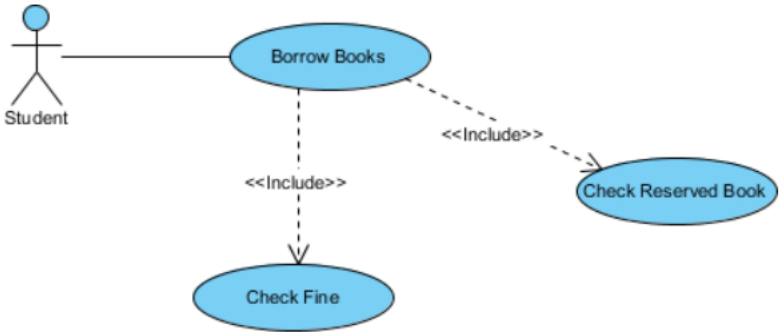


Diagrama de interacción

Se emplea para captar el comportamiento interactivo de un sistema. Los diagramas de interacción se centran en describir el flujo de mensajes dentro de un sistema y ofrecen contexto para una o más líneas de vida dentro de un sistema. Además, los diagramas de interacción pueden emplearse para representar las secuencias ordenadas dentro de un sistema, y actúan como medio para visualizar los datos en tiempo real vía UML.

Diagrama de tiempos

Se utilizan para mostrar interacciones cuando el propósito principal del diagrama es razonar sobre el tiempo. Se centran en las condiciones cambiantes dentro y entre las líneas de vida a lo largo de un eje de tiempo lineal. Describen el comportamiento de los clasificadores individuales y sus interacciones, centrándose en el momento en que ocurren los eventos que provocan cambios en las condiciones modeladas de las líneas de vida

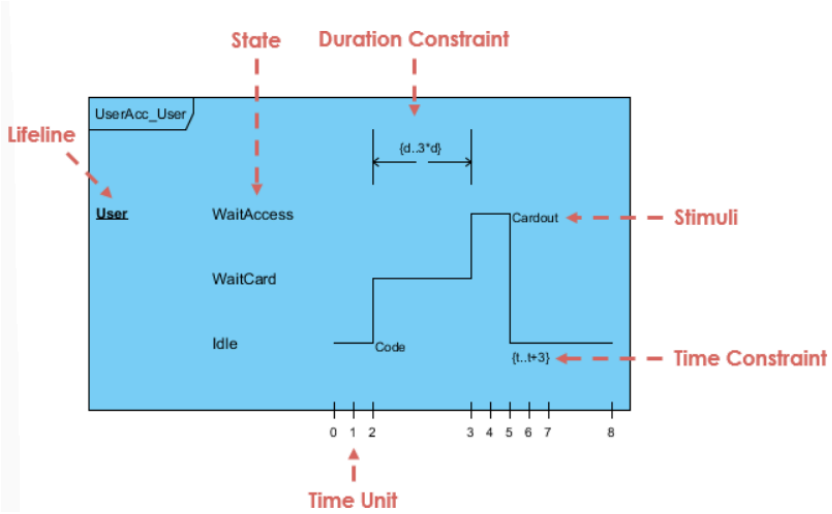
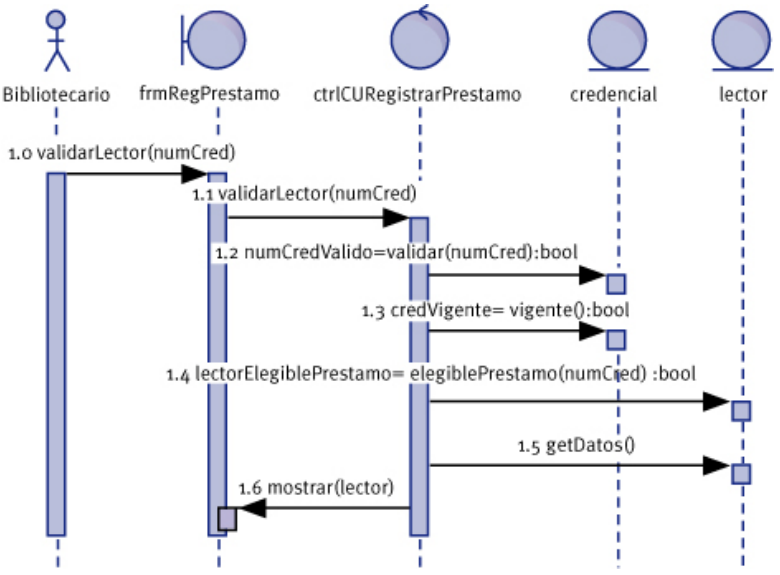


Diagrama secuencia

Representa los eventos en orden cronológico, razón por la que a veces se le llama diagrama de eventos o escenario de eventos. El orden (es decir, la secuencia exacta) es más importante que los puntos específicos en el tiempo.



Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo —y en qué orden— un grupo de objetos funcionan en conjunto. Tanto los desarrolladores de software como los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente. A los diagramas de secuencia en ocasiones se los conoce como diagramas de eventos o escenarios de eventos

Diagrama de comunicación
Representan las relaciones e interacciones entre distintos objetos de software. Enfatizan los aspectos estructurales de un diagrama de interacción y se centran en la arquitectura de los objetos más que en el flujo de los mensajes.

Un diagrama de comunicación brinda los siguientes beneficios:
Destacan cómo se conectan las líneas de vida.
Se centran en los elementos dentro de un sistema más que en el flujo de mensajes.
Proporcionan más énfasis en la organización que en la cronología.

Bibliografía

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/

https://www.lucidchart.com/pages/es/