

**1. ¿Quién debe definir los requisitos con los que se debe elaborar un programa informático?**

--El analista de sistemas es el que se encarga de definir los requisitos haciéndolo con la mayor calidad posible e intentar que coincida con las expectativas de los clientes y usuarios. Por supuesto en los proyectos informáticos debe especificarse desde el principio que debe hacer el software que queremos y que procesos debe llevar a cabo, de lo contrario, con el tiempo surgirán complicaciones. Sin las especificaciones necesarias, el tiempo y los recursos no pueden planificarse de manera realista y es muy probable que los clientes soliciten cambios a largo plazo y que quizás el presupuesto no los cubra.

**2. ¿Cuál es la diferencia entre requisito funcional RQF y requisito no funcional RQNF?**

X El requisito funcional lo especifica el usuario, mientras que el requisito no funcional lo especifica personal técnico como Arquitecto, líderes técnicos y desarrolladores de software.

X El requisito funcional es también la actividad que debe realizar el sistema; por otro lado, los no funcionales dependen de la criticidad de la aplicación. Por ejemplo, si su aplicación no es crítica y puede vivir con

tiempo de inactividad, es posible que no necesite desarrollar un código complejo de recuperación ante desastres y

conmutación por error, lo que reduce el tiempo total de desarrollo de su aplicación.

X Los requisitos funcionales definen la funcionalidad de un software, es decir, qué pueden hacer, mientras

que los requisitos no funcionales definen, otras cosas que no son requeridas por el usuario, pero requeridas por un

proveedor de servicios o el software en sí, como el registro, es un requisito no funcional para respaldar un

aplicación, no utilizada directamente por el usuario, pero esencial para solucionar cualquier problema en un entorno

de producción.

X Los requisitos no funcionales a veces se definen en términos de métricas (algo que se puede medir sobre

el sistema) para hacerlos más tangibles.

X Los requisitos no funcionales también pueden describir aspectos del sistema que no se relacionan con su

ejecución, sino más bien con su evolución en el tiempo (como mantenibilidad, extensibilidad, documentación, etc.).

CÓDIGO	RQF1
NOMBRE DEL REQUISITO	Inicio de sesión.
DESCRIPCIÓN	Los usuarios deben ingresar con datos únicos los cuales le asignaran un rol sobre la plataforma.
REQUERIMIENTO	Control de usuarios.

#### RQF2

CÓDIGO	RQF2
NOMBRE DEL REQUISITO	Contacto con administradores
DESCRIPCIÓN	Los usuarios podrán resolver inquietudes enviando un mensaje directo con los administradores.
REQUERIMIENTO	Contacto por PQR.

#### RQNF1

CÓDIGO	RQNF1
NOMBRE DEL REQUISITO	Adaptación a dispositivos
DESCRIPCIÓN	La página debe adaptarse a cada dispositivo en el que sea ejecutada, dándole una mejor perspectiva al producto.
REQUERIMIENTO	Responsividad.

#### RQNF2

CÓDIGO	RQNF2
NOMBRE DEL REQUISITO	Revisar enlaces de descarga.
DESCRIPCIÓN	Revisar que las fuentes externas de descarga sean seguras para los usuarios.
REQUERIMIENTO	Verificar fuentes.

3. ¿Qué es SCRUM? y defina 4 características importantes.

--SCRUM es un conjunto de buenas prácticas para trabajar en equipo. Se usan para aumentar la producción y mejorar la entrega del producto de una manera rápida y concorde a la solicitud del cliente.

1. Entrega temprana y continua de software con valor.

2. El desarrollo interactivo.

3. La colaboración e interacción con el cliente.

4. Adaptación.

4. ¿Qué es RUP? y defina 3 características importantes.

--Es la metodología estándar para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías que se adaptan al contexto y las necesidades de cada organización.

Características importantes

1. Modelado visual del software.
2. Verificación de la calidad del software.
3. Pretende implementar las mejores prácticas en Ingeniería de Software, de forma que se adapte a cualquier proyecto.

5. ¿Qué es UML? y defina 3 características importantes?

El lenguaje unificado de modelado (UML) es una representación gráfica común que busca facilitar la comprensión de sistemas de software o hardware complejos.

1. Permite explicar mejor visualmente un proceso
2. Ayuda a construir rápidamente una idea sobre un tema.
3. Contiene un esquema amplio de documentación que permite comunicar cada detalle.

6. ¿Qué es y qué representa un diagrama de casos de uso?

Es una secuencia de acciones que un sistema lleva a cabo que da lugar a un resultado de valor observable para un actor particular, alguien o algo fuera del sistema que interactúa con el mismo.

Este se representa de la manera en cómo un "Actor" (cliente) opera con el sistema que estamos desarrollando, y además de la forma como interactúan los elementos.

7. Partes de un caso de uso:

a. Escenario: Un escenario en UML es una circunstancia o situación, en la que se puede encontrar sistema.

b. Actor: Se especifica un rol jugado por un usuario o cualquier otro sujeto que interactúa con el sistema.

c. Tipos de relación:

- Asociación: Es una relación que describe un conjunto de vínculos entre clases, e indican la información más importante de las asociaciones. Se representa utilizando una línea continua.

- Extiende: (<<extend>>) es un modelo de dependencia. Que ofrece una forma de extensión más controlada que la relación de generalización.
- Generalización: Relación entre un caso de uso general y otro más específico que hereda características y añade otras.
- Incluye: Que indica que un caso de uso es incluido dentro de otro. Remplazando el uso común de la antigua relación **usa**.
- Realización: Establece una relación entre el caso de uso y los diagramas que describen la funcionalidad del caso de uso

d. Estados: son una secuencia de estados los cuales recorre un objeto durante su ciclo de vida y permite describir los cambios de estado de un objeto o componente.

- Elementos de un diagrama de transición de estados.
- Estado de un objeto: periodo de tiempo en la vida de un objeto en el cual está esperando la ejecución de alguna operación
- Evento: es una acción que se ejecuta de forma inesperada la cual puede causar la transición de un objeto de un estado a otro.
- Transición: Una transición se representa con una flecha continua que une dos estados y que se dirige al estado al que cambia el componente. Junto a ella se coloca una etiqueta que debe contener al menos el nombre del evento que provoca la transición.

Transición Simple

Transición Interna

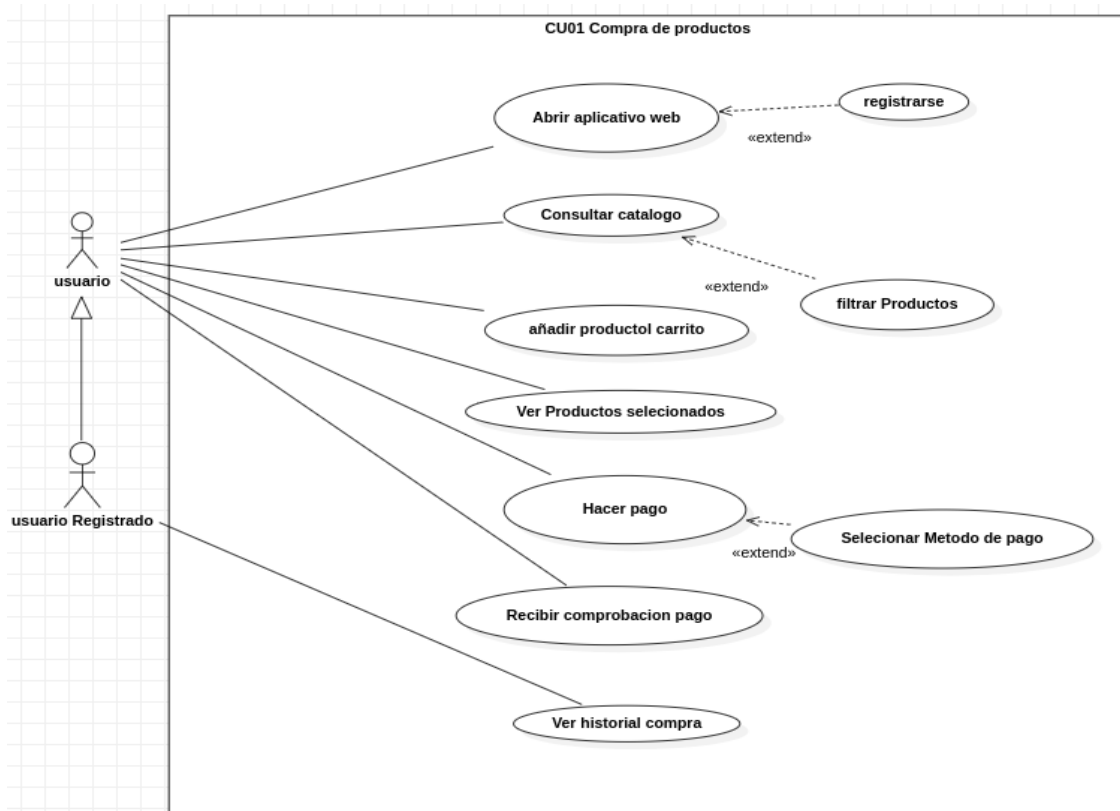
- Subestados: Un estado puede descomponerse en subestados, con intercambios de información entre ellos y conexiones al nivel superior.
- Transiciones temporizadas: son actividades con tiempo definido el cual se interrumpe cuando el evento esperado tiene lugar.

8.Componentes de las Casos de Uso. Desde su conocimiento elabore de manera individual una lista de las personas que viven con usted en su hogar. Después le va a asignar a cada una de ellas las tareas que realizan para que el hogar funcione, luego realiza el diagrama de casos de uso que permita ver el comportamiento de los actores.

9.Diseñe un diagrama de casos de uso con su respectiva ficha técnica para los usuarios de una tarea/proceso de la tienda virtual de equipos de cómputo.

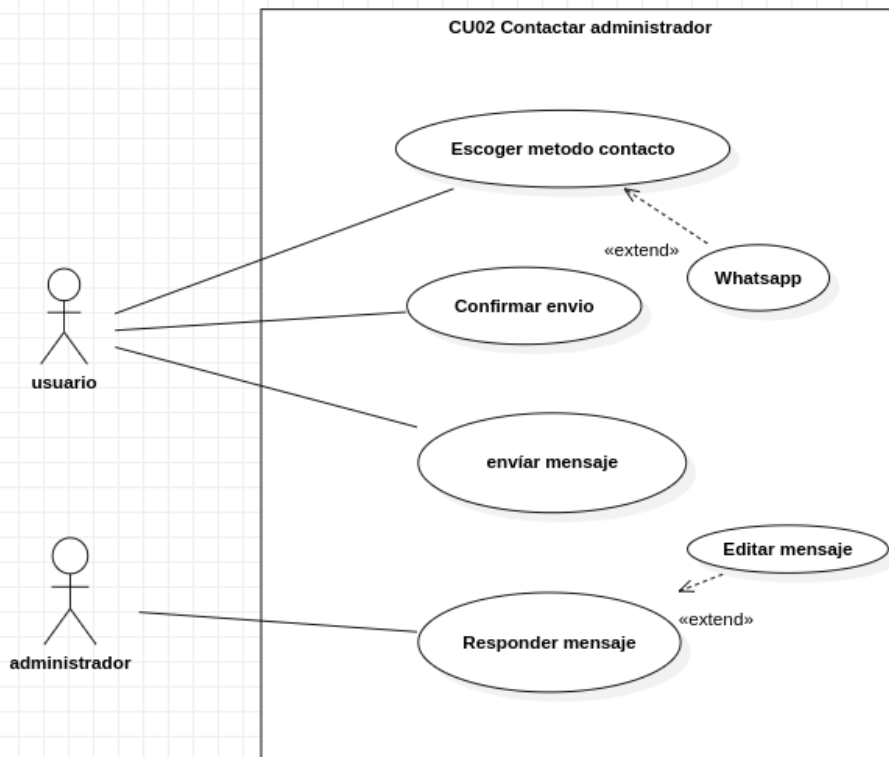
Nombre	CU01 Compra de productos
Autor	Juan David Toloza Ortega and Nefer Yesid Sánchez Gómez
Fecha	13/10/2022

<p>Descripción:</p> <p>El usuario tipo cliente podrán ver la lista de productos disponibles de la tienda y añadir al carrito de compras los productos que deseen, efectuar la compra o añadir a la lista de deseos.</p>
<p>Actores:</p> <p>usuario</p>
<p>Precondiciones:</p> <p>El usuario debe abrir el aplicativo web.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> <li>1. El sistema muestra una opción de registro en la plataforma.</li> <li>2. El usuario consulta catálogo de productos.</li> <li>3. El usuario puede filtrar los tipos de productos.</li> <li>4. El usuario agrega productos al carrito de compras.</li> <li>5. El sistema comprueba que haya al menos un producto en el carrito de compras.</li> <li>6. El sistema muestra los productos seleccionados.</li> <li>7. El usuario procede con la compra.</li> <li>8. El usuario selecciona método de pago.</li> <li>9. El sistema comprueba si el método de pago posee fondos suficientes.</li> <li>10. El usuario hace transacción bancaria.</li> <li>11. El usuario recibe aprobación de compra y se genera el envío.</li> </ol>
<p>Flujo alternativo:</p> <p>5.1 Si no hay productos en el carrito de compras, el sistema muestra un mensaje “Carrito vacío”.</p> <p>9.1 Si el método de pago no posee fondos suficientes el sistema muestra un mensaje de error. Luego ir al paso 8.</p> <p>11. Si el usuario está registrado en la plataforma, podrá acceder a el registro de compras realizadas.</p>
<p>Postcondiciones:</p> <p>1. El sistema muestra un mensaje “Gracias por su compra” y envía al usuario al paso 2.</p>



Nombre	CU02 Contactar Administrador
Autor	Juan David Toloza Ortega and Nefer Yesid Sánchez Gómez
Fecha	13/10/2022
Descripción: El usuario podrá ponerse en contacto con el administrador del aplicativo web, para resolver dudas o resolver algún problema con el software.	
Actores: Usuario, administrador.	
Precondiciones: El usuario debe tener email o número de teléfono.	
Flujo normal: <ol style="list-style-type: none"> <li>1. El usuario va a la sección de contacto.</li> <li>2. El usuario escoge el método en que se quiere contactar con el administrador.</li> <li>3. El usuario diligencia el formulario.</li> <li>4. El usuario confirma el envío del mensaje y sus datos.</li> <li>5. El sistema revisa que no queden casillas en blanco.</li> <li>6. El sistema muestra un mensaje de enviado en pantalla.</li> <li>7. El sistema pone en lista de espera el mensaje del usuario.</li> </ol>	
Flujo alternativo: <p>2.1 Si el usuario escoge otro método de contacto, el sistema debe pedir los mismos datos que por los otros métodos.</p> <p>4.1 si el usuario no confirma el envío, podrá volver al paso 3 y editar el mensaje.</p> <p>5.1 Si hay casillas en blanco el sistema debe mostrar un mensaje de error, junto con las casillas vacías.</p>	
Pos condiciones:	

El administrador responde el mensaje del usuario por el mismo método de contacto.



10. ¿Qué es y para qué sirve un diagrama de clases?

Los diagramas de clase son un tipo de diagrama UML muy útil ya que describen detalladamente lo que está presente en el sistema que se está modelando. Además, que permite trazar claramente la estructura concreta de un sistema modelando sus clases, atributos, operaciones y relaciones entre objetos

11. ¿Qué es y cómo se aplica la visibilidad de los campos y la multiplicidad en los diagramas de clases?

La visibilidad o accesibilidad determina el alcance que tiene un atributo u operación.

El enlace (scope) representa qué partes pueden acceder a dicho miembro y se compone de los siguientes modificadores de visibilidad.

“público” que se representa con el signo +

“privado” que se representa con el signo -

La multiplicidad es una restricción que se pone a una asociación y limita el número de instancias de una clase que pueden tener esa asociación con una instancia de otra clase.

Las multiplicidades más comunes son las siguientes:

- 0..1. Sin instancia o sólo una (opcional)
- 1. Una y siempre una instancia (obligatorio)
- \*. Cero o múltiples instancias (opcional)
- 1..\*. Múltiples instancias, pero al menos una (obligatorio)

Se entiende;

- Si el límite inferior es 0, la asociación es opcional. Si el límite inferior es 1 o superior implica que la asociación es obligatoria y como mínimo han de ocurrir ese número de instancias.
- El límite superior puede estar acotado por un número exacto o ser indefinido. En este último caso se usa el símbolo \* (asterisco).
- Si ambos límites coinciden, se colapsan en un único número. Por ejemplo, 1..1 pasa a ser 1.

12. ¿Qué tipos de relación se pueden presentar en un diagrama de clases?

- Asociación.

Representan la unión entre objetos o instancias de clases y es el tipo de relación, es más general, Y muestra una dependencia semántica.

Se compone de elementos adicionales.

- Rol
- multiplicidad
- Herencia.

Le permite incorporar atributos y métodos de otras clases a una clase, el cual se añade a lo que la clase ya posee.

- Agregación.

La agregación es un tipo de relación jerárquica entre un objeto que representa la totalidad de ese objeto y las partes que lo componen. Y por lo tanto los objetos hacen parte de otro objeto completo

- Composición.

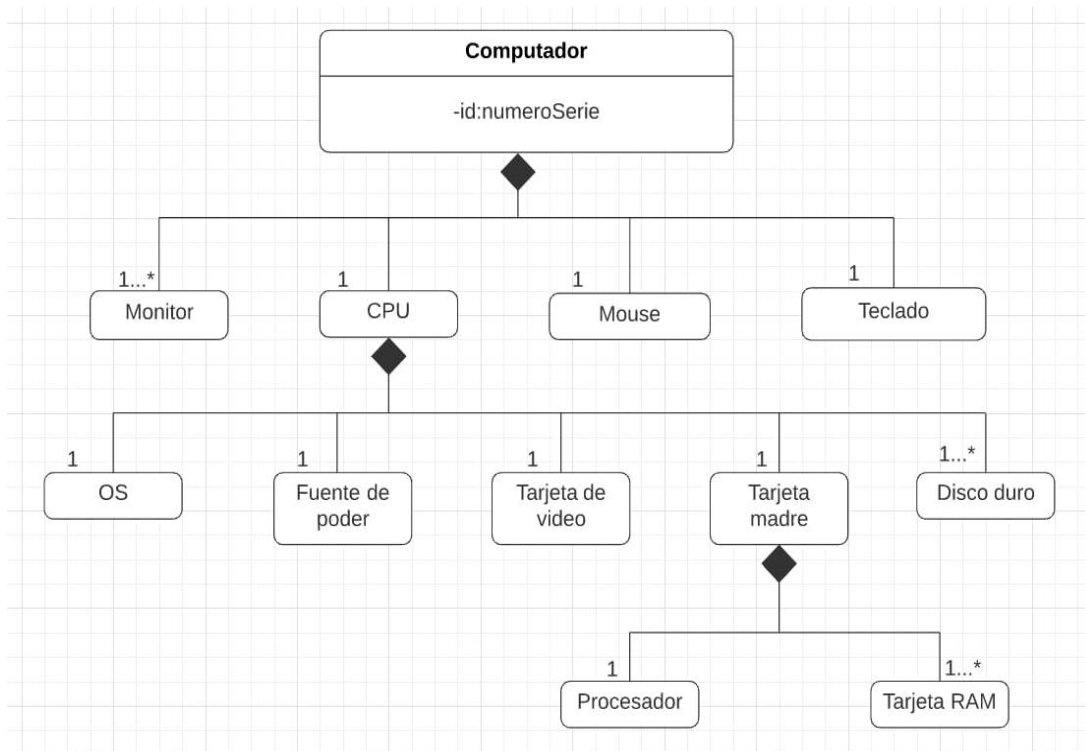
En esta relación de propiedades es más fuerte, y también coinciden la en tiempo de funcionalidad del objeto y las partes que lo componen.

- Dependencia.

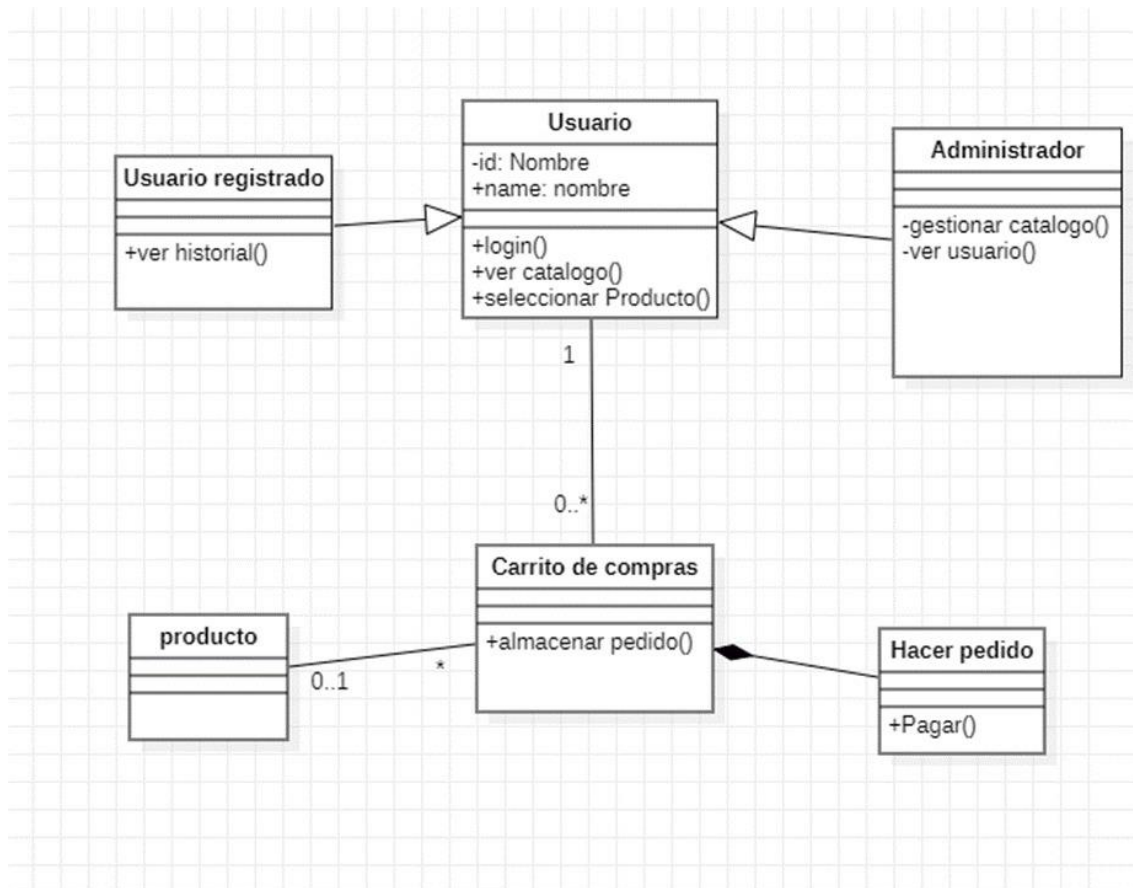


Una relación de dependencia se utiliza entre dos clases o entre una clase y una interfaz, e indica que una clase requiere de otra para proporcionar alguno de sus servicios.

**13.** Teniendo en cuenta la imagen de este enlace, donde se muestra un diagrama de clases jerárquico de “Animal, elabore un diagrama de clases teniendo en cuenta el tema de arquitectura de un computador explicando lo tipos de relación del ejemplo.



Tienda virtual



14. ¿Qué es y que representa un diagrama de secuencias?

Es un diagrama de interacción que describe cómo y en qué orden un grupo de objetos funciona en conjuntos y en también se le conoce se le conoce como (diagrama de eventos o escenario de eventos).

- Representa el los detalles de un caso de uso en UML
- Sirve para modelar la lógica de una operación, una funcion o un procedimiento sofisticado.
- Observa como los objetos y los componentes interactúan entre sí para completar un proceso.
- Busca planificar y comprender la funcionalidad detallada de un escenario actual o futuro.

15. Partes de un diagrama de secuencias:

a. Línea de vida: son una solución de modelado dinámico popular en UML porque se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.

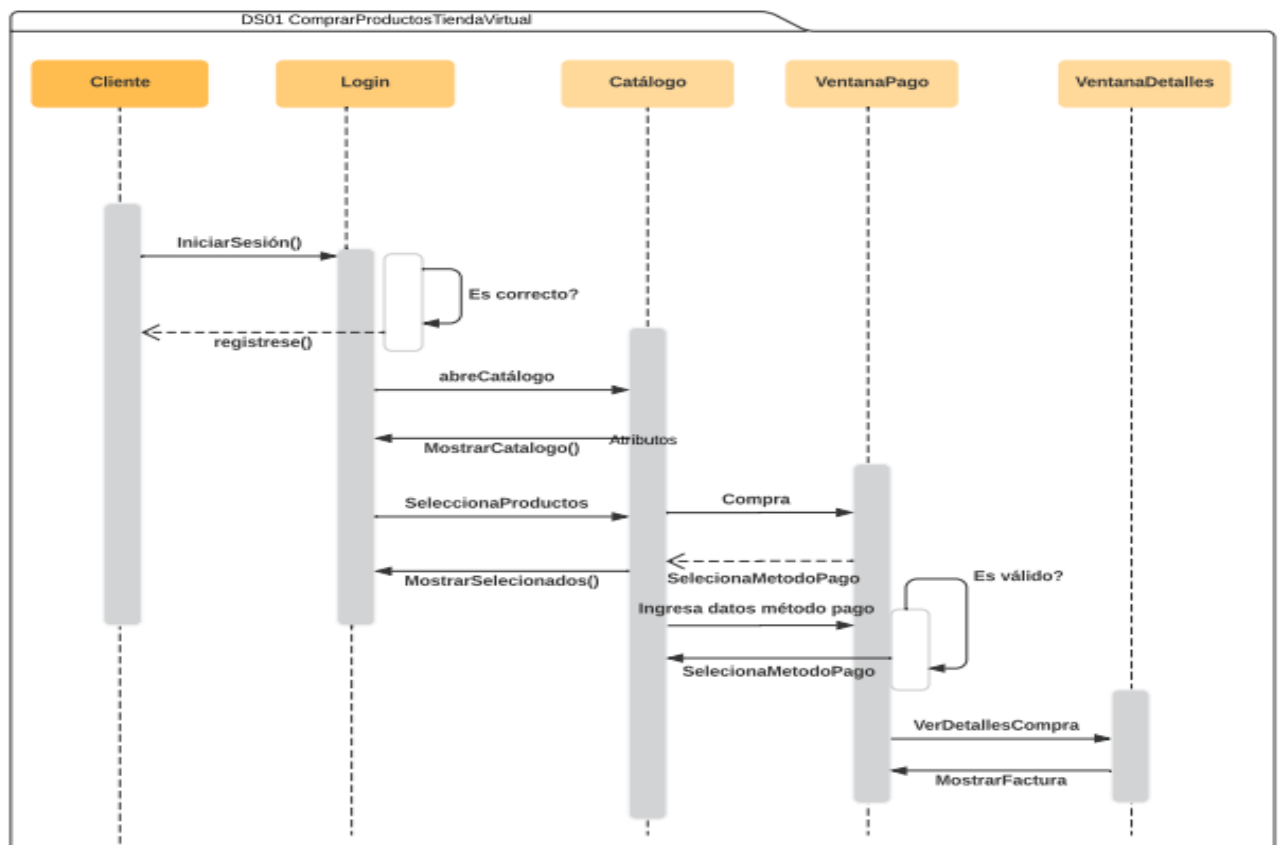
b. Ocurrencia: las ocurrencias de interacción (interaction occurrence) remiten de un diagrama a otro. Esta puerta abre la conexión en el borde exterior de la ocurrencia de interacción para los mensajes desde la interacción a la que se refiere la ocurrencia de interacción. De este modo, la puerta tiene una asociación a la ocurrencia de interacción y acepta mensajes entrantes y salientes.

c. Mensajes: es una línea que termina con una punta de flecha. Esta pasa de una línea de vida de un objeto a otra.

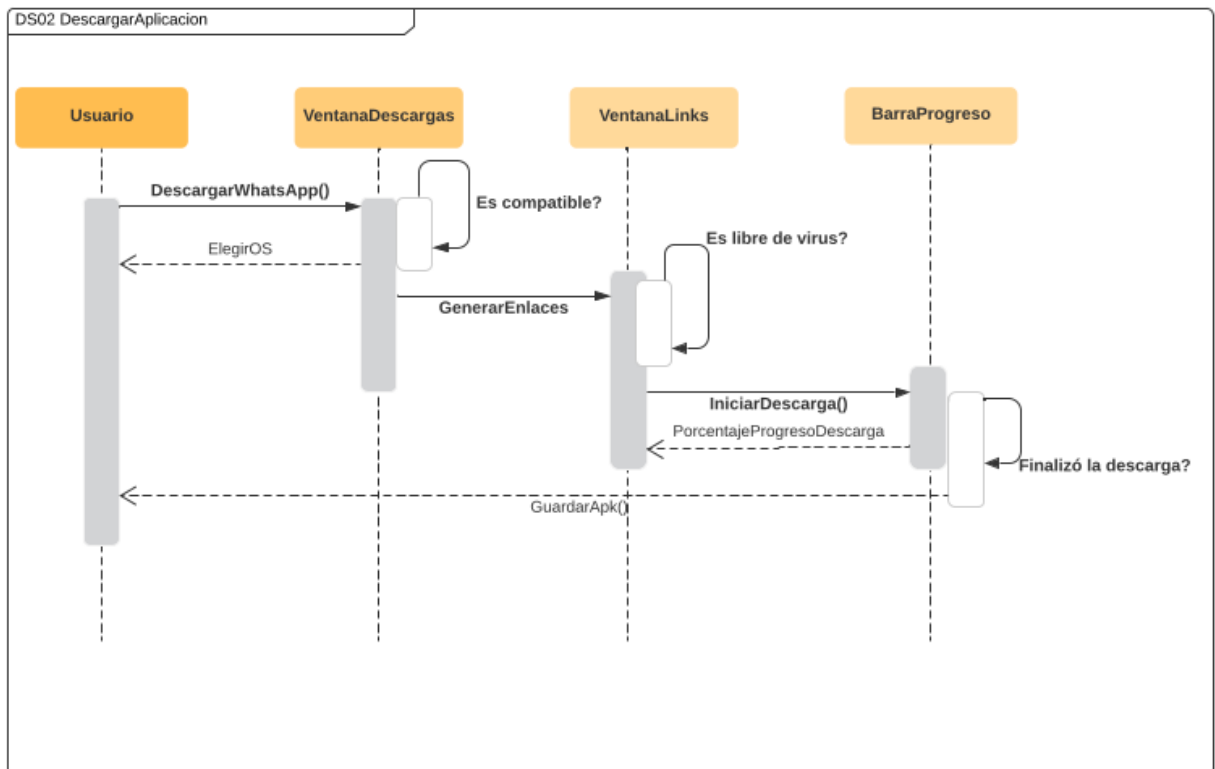
Se compone de:

- Mensaje Simple: que representa la transferencia de un control a otro.
- Mensaje Síncrono: si este mensaje es requerido es debido a que se necesita una respuesta antes de continuar.
- Mensaje Asíncrono: es un mensaje que no requiere una respuesta antes de que el remitente continúe y se utiliza para representar la comunicación entre subprocesos diferentes o la creación de un nuevo subproceso.

**16.** Diseñe un diagrama de secuencias para una tarea/proceso de la tienda virtual de equipos de cómputo.



De nuestro proyecto



17. ¿Qué es y que representa un diagrama de estados DTE?

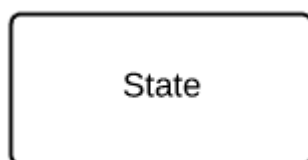
Los diagramas de estado muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación en respuesta a eventos.

El diagrama de estados muestra el comportamiento de un sistema con la máxima precisión y es la representación gráfica de los procesos.

18. Partes de un diagrama de estados:

a. Estados:

Rectángulo de esquinas redondeadas que muestra la naturaleza actual de un objeto.



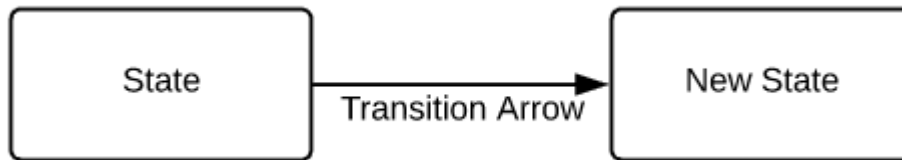
b. Subestados: Un estado contenido dentro de la región de un estado compuesto.

c. Eventos:

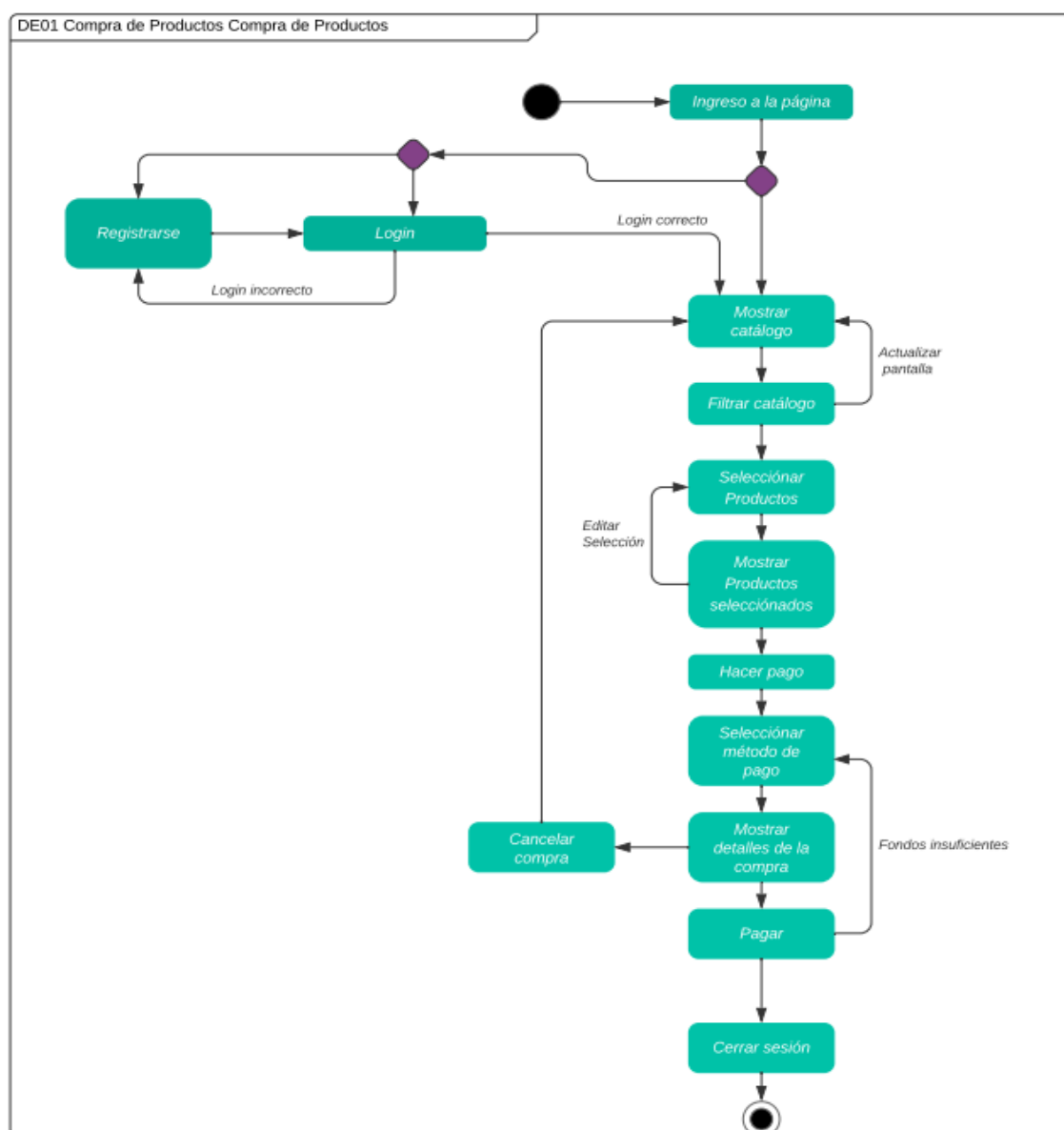
un evento es la ocurrencia de un estímulo que puede desencadenar una transición de estado.

d. Transiciones:

Una flecha que corre de un estado a otro, que indica un estado cambiante.



19. Diseñe un diagrama de estados para un proceso de una tienda virtual de equipos de cómputo.



De nuestro proyecto

