

Guía Examen Final – Análisis y Modelación de Sistemas de Software

1.- ¿Qué es el proceso unificado?

R= es más que una serie de pasos que, si se siguen, resultarán en la construcción de un sistema de información, es una metodología iterativa y por incrementos.

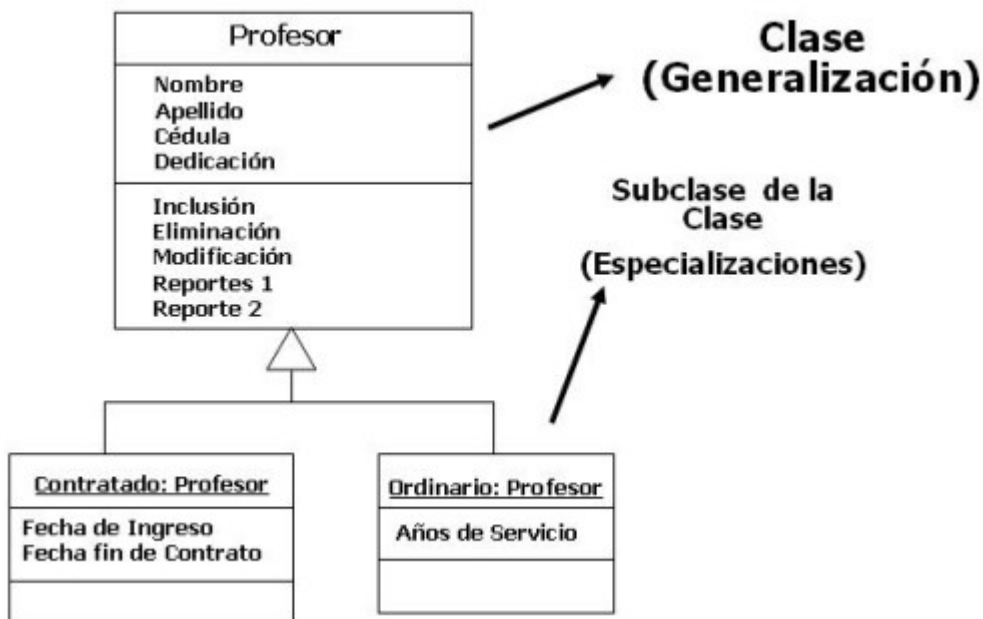
2.- ¿En qué consiste el ocultamiento de información?

R= es una manera de mejorar la capacidad de mantenimiento al volver invisibles los detalles de la implementación fuera de un objeto.

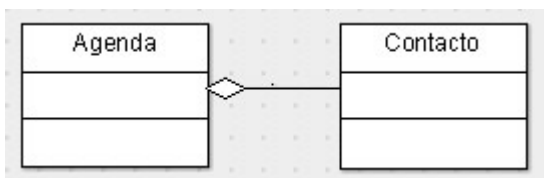
3.- ¿Qué tipo de relaciones existen entre clases en el diseño UML?

R= Generalización, agregación y asociación. Las cuales dentro de su notación como la siguiente:

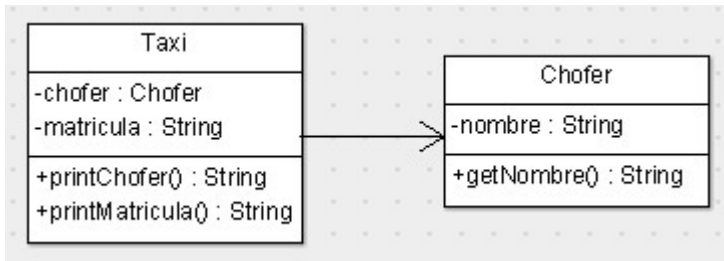
- Generalización: Un ejemplo de esto es herencia y se denota como muestra la siguiente imagen.



- Agregación: Este tipo de agregación es del tipo este objeto *se compone de* las clases Y y z. Ejemplificando lo anterior:



- Asociación: es una relación arbitraria entre dos clases.



4.- ¿Qué es un modelo?

R= es un conjunto de diagramas UML que representan uno o más aspectos del sistema de información que queremos desarrollar.

5.- ¿Qué es UML?

R= es una herramienta que usamos para representar (modelar) el sistema de información objetivo.

6.- ¿En qué consiste un workflow?

R= consta de una serie de pasos, con la finalidad de tener un modelo UML exacto del sistema que quieren desarrollar.

7.- ¿Cuál es la tarea principal del workflow de los requisitos?

R= trabajar con los clientes y con los futuros usuarios del sistema de información para determinar qué quieren.

8.- ¿En qué consiste el dominio de aplicaciones?

R= el entorno de negocios específico en el cual va a operar el sistema de información.

9.- ¿Qué es un modelo de negocios?

R= se usan los diagramas UML para describir los procesos de negocios del cliente y determina cuáles son los requisitos del cliente.

10.- ¿En qué consiste la obtención de requerimientos?

R= consiste en el proceso de descubrir los requerimientos del cliente.

11.- ¿En qué consiste el análisis de los requisitos?

R= consiste en el proceso de refinar y extender los requisitos del cliente.

12.- ¿Qué es el modelo de negocios inicial?

R= es una descripción de los procesos de una empresa. Esto se tiene que hacer con la finalidad de que proporcione una comprensión de los negocios del cliente con un todo.

13.- Menciona algunos ejemplos del cómo obtener los requerimientos.

R= Algunos ejemplos son:

- Entrevistas.
- Cuestionarios.
- Formularios.
- Focus Groups.
- Observación directa.

14.- ¿Qué es un caso de uso?

R= es un modelo de una interacción entre el sistema de información y los usuarios de éste (actores).

15.- ¿Qué es un requisito funcional?

R= es aquel que especifica una acción que el sistema de información debe ser capaz de realizar.

16.- ¿Qué es un requisito no funcional?

R= especifica las propiedades del sistema de información mismo, como las restricciones de plataforma, tiempos de respuesta, fiabilidad, entre otros.

17.- ¿Cuáles son los dos propósitos del workflow del análisis?

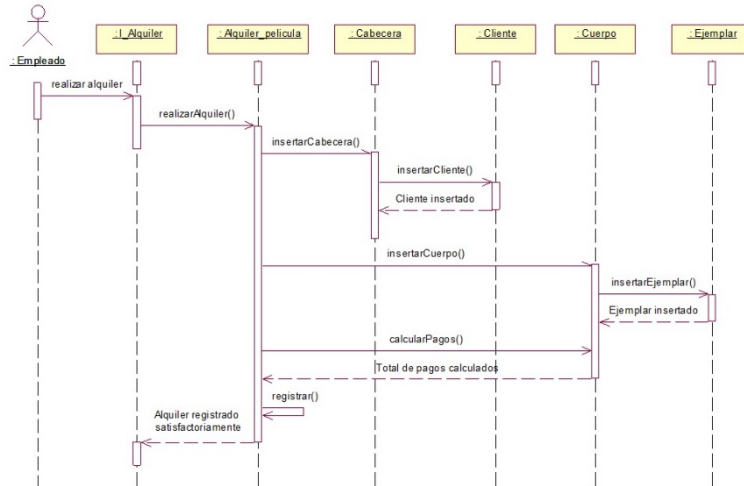
R= Primero: obtener una comprensión profunda de los requisitos; segundo, se quiere describir los requisitos de una manera fácil de mantener y que proporcione apoyos para la comprensión de la estructura del sistema que se va a desarrollar.

18.- ¿Cómo se realiza la extracción de las clases entidad?

R= este proceso se realiza en 3 pasos de manera iterativa:

1. Modelado funcional: presentar los escenarios de todos los casos de uso.
2. Modelado de clases: determinar las clases entidad y sus atributos.
3. Modelado dinámico: determinar las operaciones realizadas por o con cada clase o subclase de entidad.

19.- Diagrama de secuencia

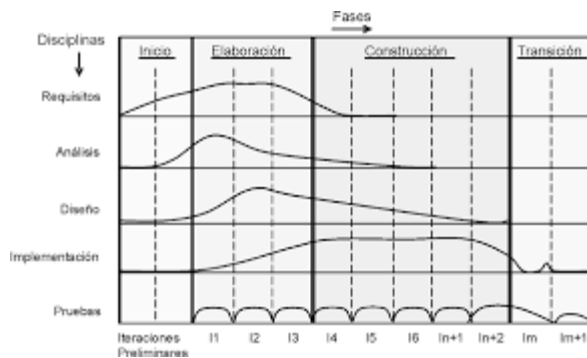


20.- ¿Cuáles son los diferentes workflows del proceso unificado?

R= existen 5 workflows dentro del proceso unificado las cuales son:

1. Workflow de requerimientos.
2. Workflow de análisis.
3. Workflow de diseño.
4. Workflow de la implementación.
5. Workflow de pruebas.

Como lo muestra la siguiente figura:



21.- ¿Cuáles son las fases que conforman el proceso unificado?

R= Las fases que existen dentro del proceso unificado:

1. Fase de iniciación: es determinar si vale la pena desarrollar el sistema de información deseado.
2. Fase de elaboración: su objetivo es refinar los requisitos iniciales y la arquitectura, monitorear los riesgos y refinar sus prioridades, caso de la empresa y producir un plan de administración del proyecto.

3. Fase de construcción: su propósito es producir la primera versión operativa de calidad del sistema de información, a veces llamada versión beta.
4. Fase de transición: su objetivo es asegurar que los requisitos del cliente se hayan cumplido.

22.- ¿A qué se refiere cuando se habla de verificación?

R= se refiere al proceso de determinar si un workflow específico se ha realizado de manera correcta.

23.- ¿A qué se refiere cuando se habla de validación?

R= es el proceso de evaluación intensivo que ocurre justo antes de que el sistema de información se entregue al cliente.

24.- ¿Qué implica cuando se habla de calidad?

R= la calidad implica excelencia de algún tipo. Basado en esto, la calidad de un sistema de información es el grado en el cual el sistema cumple con sus especificaciones.

25.- ¿Cuál es la diferencia entre defecto y falla?

R= un defecto es la terminología estándar del IEEE para lo que comúnmente se conoce como error(bug), mientras que una falla es el comportamiento incorrecto observado en el sistema de información como una consecuencia del defecto.

26.- ¿Qué es un error?

R= un error es una equivocación que cometió el programador.

27.- ¿Cómo se realiza una inspección?

R= Una inspección consta de 5 pasos para realizarla:

1. Descripción general del artefacto que se va a inspeccionar.
2. Preparación: los participantes tratan de comprender el artefacto a detalle.
3. Inspección: un participante revisa paso por paso el artefacto junto con el equipo de inspección, asegurándose de que cada elemento esté cubierto y que cada alternativa se considere.
4. Adaptación: el individuo responsable de ese artefacto resuelve todas las fallas y problemas observados en el informe escrito.
5. Seguimiento: el moderador debe asegurar que cada problema individual que haya surgido se resuelva de manera satisfactoria, ya sea reparando al artefacto o esclareciendo qué elementos se etiquetaron como fallas de manera incorrecta.

28.- Explica en qué consiste un caso de prueba de caja negra

R= Son los casos de prueba que se preparan meramente con base en las especificaciones.

29.- Explica en qué consiste un caso de prueba de caja blanca o de cristal

R= son un conjunto de casos de prueba que, cuando se ejecuten, aseguren que cada línea de código se ejecute al menos una vez.

30.- ¿Qué deben de evaluar las pruebas basadas en ejecución?

R= Estas pruebas deben de evaluar:

- Utilidad: grado en el cual el sistema de información satisface las necesidades del usuario.
- Fiabilidad: medida de la frecuencia y la gravedad de sus fallas, o sea, es necesario saber con qué frecuencia falla el sistema.
- Solidez o Robustez: implica una serie de factores, como la gama de condiciones de operación, la posibilidad de obtener resultados inaceptables como una entrada válida y la capacidad de aceptación de los efectos cuando el sistema de información recibe una entrada inválida.
- Desempeño: saber el grado en el cual el sistema de información cumple con las restricciones con respecto al tiempo de respuesta o a los requisitos de espacio.
- Corrección: un sistema de información es correcto cuando cumple con todas sus especificaciones.

31.- ¿Cuándo se deben de detener las pruebas?

R= las pruebas se deben de detener cuando el sistema de información es destituido y retirado del servicio.

32.- ¿Cuáles son los criterios importantes a la hora de implementar una Interfaz de Usuario?

R= Dentro de los criterios a considerar a la hora de la implementación de la interfaz de usuario están los siguientes:

- Tener en cuenta los dispositivos de entrada y salida.
- La consistencia en el diseño de interfaces. Se tiene que establecer una plantilla de cómo serán las interfaces y que si el botón de *Atrás* aparece en el lado derecho siga apareciendo en el lado derecho.
- Tener contemplado la posibilidad de recuperarse de un error cometido por el usuario, hizo click donde no debía si está bien diseñada le debe de permitir recuperarse a un estado anterior.
- El uso de metáforas como la palabra *menú*, *drag and drop*, *carpetas*, *ventanas*, etc.

33.- Otros diagramas:

Casos de uso:

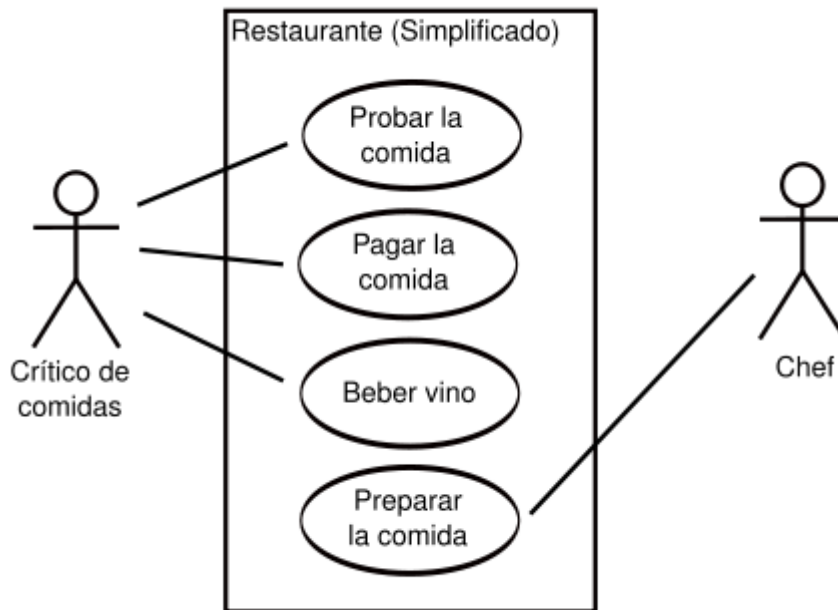


Diagrama de clase de análisis: recordar que el diagrama de clase de análisis simplemente consiste en las clases del dominio del problema junto con sus atributos, **NO** se deben incluir los métodos que tendrá la clase como `get()`, `set()`, etc.

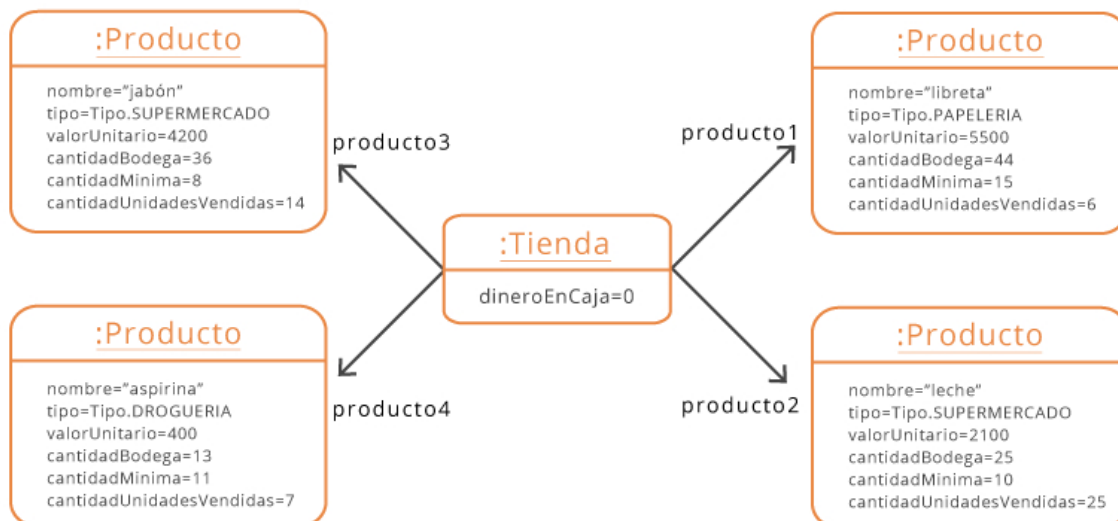


Diagrama de clase de diseño: este diagrama contendrá las clases dominio del problema, así como aquellas que salieron después del análisis del diagrama de

clase de análisis. En este diagrama si se incluyen los métodos con los que dispondrá cada clase, como se muestra en la siguiente imagen.

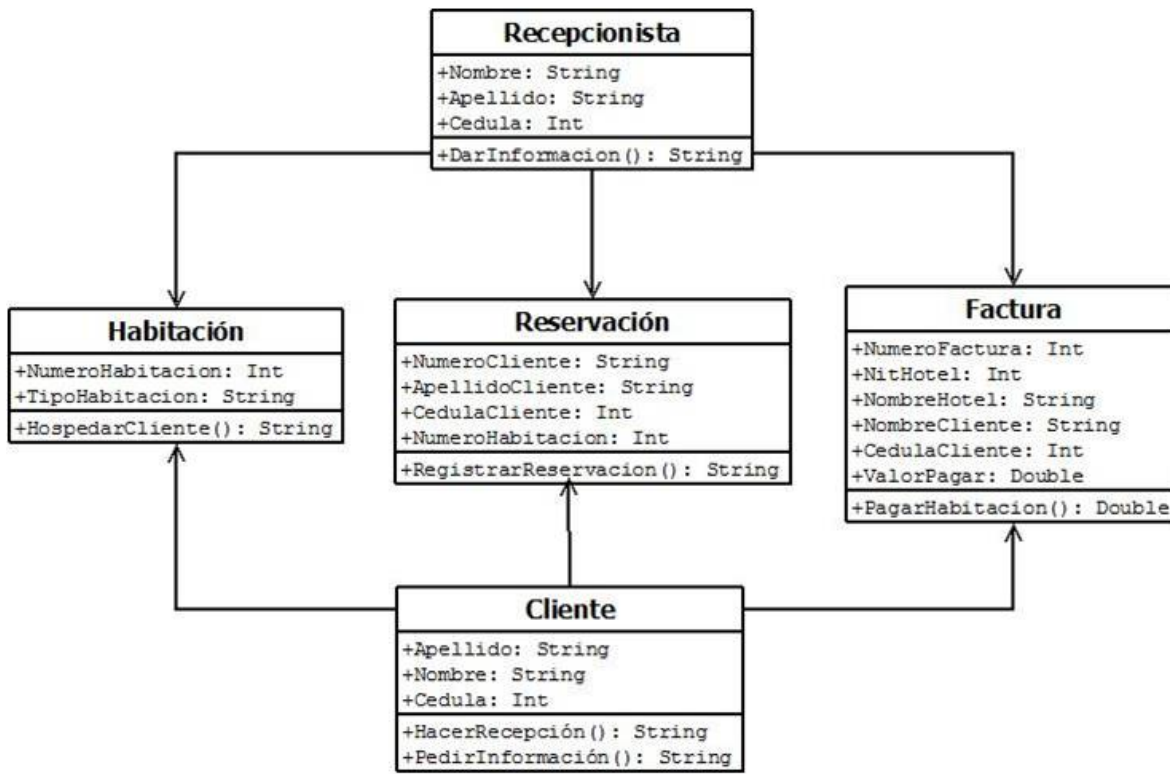


Diagrama de secuencia: en este diagrama se pondrá la secuencia del cómo interactúan los actores con las clases del sistema, recordar la notación UML para que el diagrama este realizado de manera correcta, recordar que la línea de conexión debe de tener el método con el cual se comunican las clases entre ellas, actores entre actores o actores y clases. Como se muestra en la imagen:

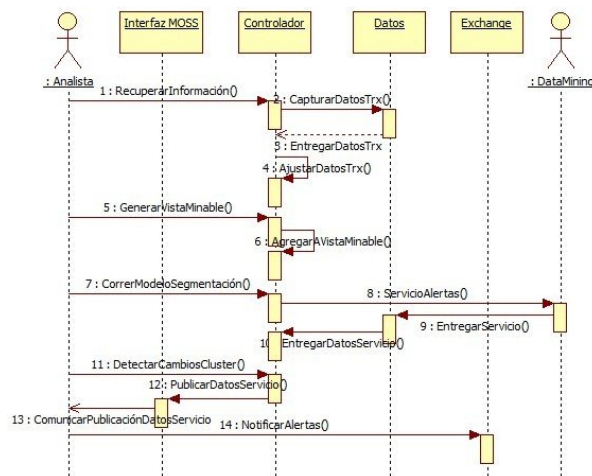


Diagrama de estado: en este diagrama se muestra la secuencia de estados que pasa un objeto durante su vida. Debido a un input el objeto cambiará en un estado

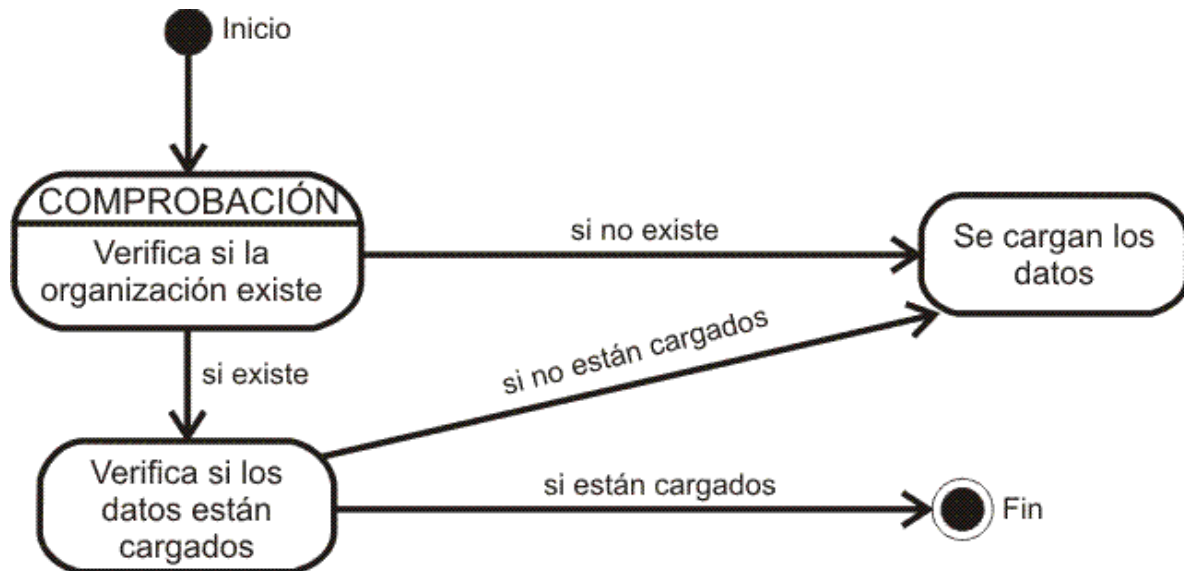
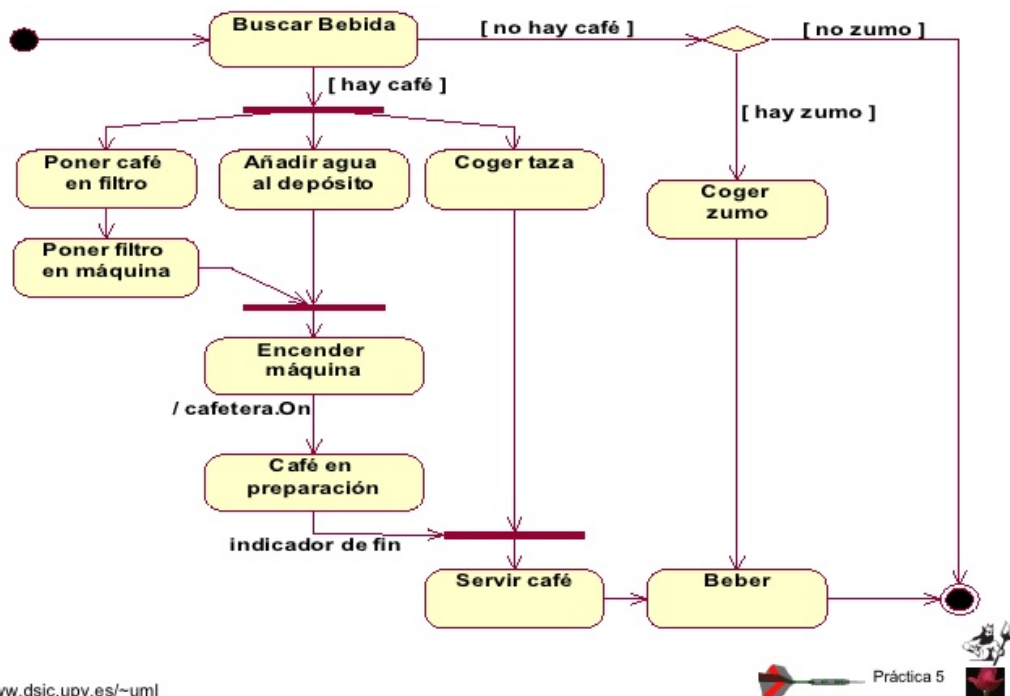


Diagrama de actividad: puede mostrar una secuencia de acciones o actividades ejecutadas por un sistema, puede contener bifurcaciones y bucles como el que se muestra en la imagen y este diagrama puede ser separado con swimlines.

Diagrama de actividad sin swimlines:



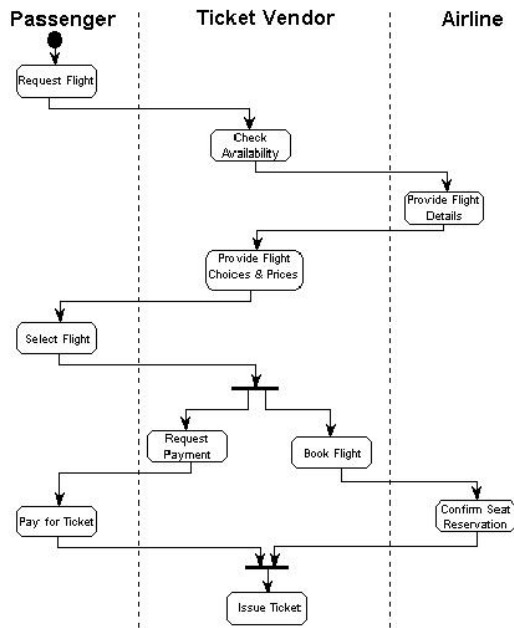


Diagrama entidad-relación: en este diagrama se ponen las llaves primarias y se relacionan entre ellas usando la notación UML como se muestra en la imagen.

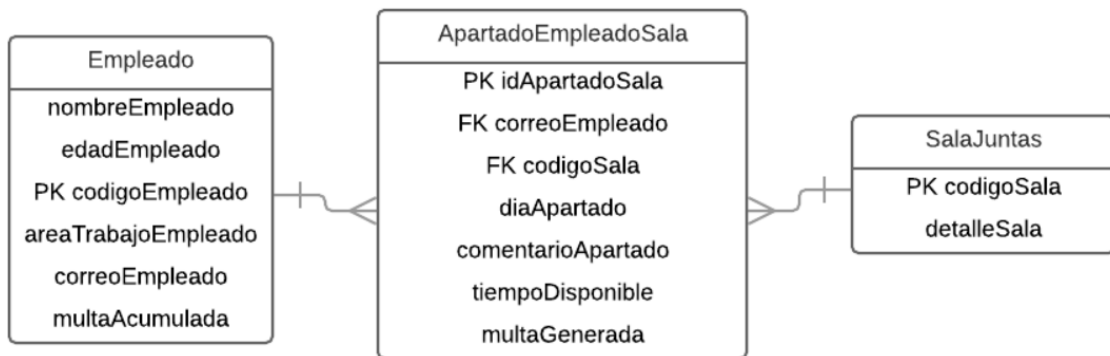


Diagrama de despliegue: un diagrama de despliegue muestra en que componentes de hardware el software está instalado. Como se muestra en la imagen.

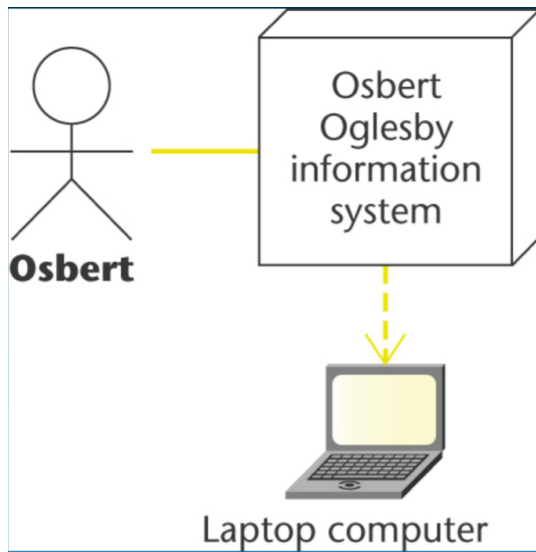


Diagrama de componentes: representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre esos componentes, como se muestra en la siguiente imagen.

Aplicación Almacén Deportes

