



UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

**FACULTAD DE INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN Y TELECOMUNICACIONES**

PRACTICO # 4

NOMBRE: MOGIANO GUTIERREZ MOISES LEONARDO

NRO. REGISTRO: 218034121

CARRERA: ING. DE SISTEMAS

DOCENTE: MSC.ING. ANGÉLICA GARZÓN CUÉLLAR

MATERIA: SISTEMAS DE INFORMACIÓN 1-INF342-SA

FECHA: 14/12/2021

PRACTICO # 4

1. Como se determina los requerimientos de información para desarrollar un sistema?

Determinar requerimientos consiste en estudiar un sistema para conocer cómo trabaja y donde es necesario efectuar mejoras.

2. Cuáles son los elementos de un sistema de información?

Los elementos de un sistema son:

- **Entradas:** Son los ingresos del sistema que pueden ser recursos materiales, recursos humanos o información.
- **Proceso:** Es lo que transforma una entrada en salida.
- **Salidas:** Son los resultados que se obtienen de procesar las entradas. Son el propósito para el cual existe el sistema.

3. Cuál es el papel del analista de sistema?

El analista encuentra los actores y casos de usos, a su vez estructura los modelos de casos de usos.

4. Determinación de la factibilidad y el manejo de las actividades de análisis y diseño?

Por medio del estudio de factibilidad los analistas de sistemas recopilan datos que permiten a la administración decidir si continúan con un estudio de sistema completo. La planeación del proyecto incluye la estimación del tiempo requerido por cada una de las actividades del analista, su calendarización y la agilización de ellas, si es necesario para asegurar que un proyecto sea terminado a tiempo. Una técnica de que dispone el analista de sistemas para la calendarización de tareas es la gráfica de Gantt, que despliega actividades en forma de barras en una gráfica.

5. Explique las diferentes metodologías que existen para desarrollar software?

Las principales metodologías tradicionales o clásicas son:

- **Waterfall (cascada):** Es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden.

Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.

-Prototipado: Se basa en la construcción de un prototipo de software que se construye rápidamente para que los usuarios puedan probarlo y aportar feedback. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo que se basa en el método de prueba y error para comprender las especificidades del producto.

-Espiral: Es una combinación de los dos modelos anteriores, que añade el concepto de análisis de riesgo. Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología da nombre a su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto.

-Incremental: En esta metodología de desarrollo de software se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías.

-Diseño rápido de aplicaciones (RAD): Esta metodología permite desarrollar software de alta calidad en un corto periodo de tiempo. Los costes son mucho más altos y el desarrollo más flexible, aunque requiere una mayor intervención de los usuarios. Por otro lado, el código puede contener más errores, y sus funciones son limitadas debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces para conseguir una aplicación completa de forma rápida.

6. Describa la diferencia que existe entre el análisis y diseño estructurado, orientado a objeto y uml-puds

Tabla de Diferencias

Análisis y Diseño Estructurado	Análisis y Diseño Orientado a Objetos
Se consideran los elementos o perspectivas básicas del análisis (Entrada-Proceso-Salida), en función del Software.	Se consideran los conceptos básicos como el Objeto y el Atributo, el todo y sus partes (software), clases y miembros. Modela los objetos que son parte de él.
Utiliza el diagrama estructurado como representación gráfica del sistema.	Utiliza el diagrama orientado a objetos como representación gráfica del sistema.
Consta de 5 Fases (Análisis, Diseño, Codificación, Pruebas e Integración).	Consta de 4 Fases (Análisis, Diseño, Evolución y Modificación).
No enfoca apropiadamente el diseño de familias de programas. Asume una progresión relativa uniforme de pasos de elaboración.	Une a los usuarios y a los diseñadores. Permite proporcionar una descripción completa del problema, legible y revisable por las partes interesadas y verificables contra la realidad.
No acomoda el tipo de desarrollo evolutivo. No enfoca los posibles modos futuros de desarrollo de software.	Si están correctamente definidas las jerarquías de clase, hacer modificaciones no es tan costoso como en el caso de programación tradicional. Sólo hay que entrar en la parte de Evolución para hacer modificaciones.
El Diseño inicia una vez que ha culminado la fase de análisis de sistema.	El Diseño inicia aún antes de concluir con la etapa de análisis. Se recomienda analizar un poco y diseñar. Esta etapa debe concluir una vez que se establecieron claves y mecanismos importantes.
En este análisis se llega solo a la fase de integración y no toma en consideración los cambios que ocurren dentro del sistema en el proceso de análisis y diseño de sistemas.	Un programa que se usa en un ambiente real necesariamente debe cambiar. Los cambios difieren un poco de los requeridos en evolución, pues contemplan la introducción de nuevas funcionalidades no previstas en el problema original.
Las herramientas utilizadas son: Diagrama de Flujo de Datos, Diagramas de Entidad-Relación, Diagrama de Transición de Estados.	Las herramientas utilizadas son: Diagramas de Clases, Diagrama de Objetos, Diagramas de Módulos, Diagramas de Procesos, Diagramas de Transición de Estados, Diagramas de Tiempo.

El análisis está orientado a los Procesos del sistema.	El análisis está orientado a los Objetos.
--	---

7. Define el concepto de UML y explique ampliamente su filosofía de trabajo?

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software orientada a objetos.

UML usa las fortalezas de estos tres enfoques para presentar una metodología más uniforme que sea más sencilla de usar. UML representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios.

8. Cuantos tipos de prototipos se incorporan en la metodología del ciclo de vida?

Los tipos de prototipos que se incorporan en la metodología del ciclo de vida son 3 y son:

-Prototipo evolutivo: Realiza la construcción de una implementación parcial que cubre los requisitos conocidos, este tipo de prototipo va incorporando paulatinamente los que se van aprendiendo por el mismo desarrollo al sistema.

-Prototipo rápido: Éste no modifica el flujo del ciclo de vida y reduce el riesgo de construir proyectos que no satisfacen las necesidades de los usuarios, de tal manera que así reduce costos y aumenta en gran parte a que el proyecto a entregar sea exitoso.

-Prototipo operacional: Mezcla los dos prototipos: el evolutivo y el rápido dónde obtiene resultados rápidos sin perder calidad si se aplican bien en sistemas de gran escala.

9. ¿Qué técnicas se utilizan para recabar requisitos en el sistema, explique brevemente cada una?

Algunas de las técnicas que se utilizan para recabar requisitos en el sistema son:

-Entrevistas: La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades.

-Desarrollo Conjunto de Aplicaciones (JAD): Es una técnica que se utiliza para promover la cooperación y el trabajo en equipo entre usuarios y analistas.

Consiste en realizar sesiones en las que participan usuarios expertos del dominio junto a analistas de software.

-Desarrollo de Prototipos: Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida.

-Observación: Este método permite observar la forma en que se llevan a cabo los procesos y, por otro, verificar que realmente se sigan todos los pasos especificados.

-Estudio de documentación: Varios tipos de documentación, como manuales y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones.

-ETHICS (Implementación Efectiva de Sistemas Informáticos desde los puntos de vista Humano y Técnico): Un sistema no tiene éxito si no se ajusta a los factores sociales y organizacionales que rigen a la empresa. Se busca la satisfacción de los empleados en el trabajo a través de estudios integrales. Los requisitos técnicos del sistema serán los necesarios para mejorar la situación de los empleados (y, por lo tanto, su productividad) en función de dichos análisis.

-Tormenta de ideas (Brainstorming): Consiste en reuniones con cuatro a diez personas donde como primer paso sugieren toda clase de ideas sin juzgar su validez; por muy disparatadas que parezcan, y después de recopilar todas las ideas se realiza un análisis detallado de cada propuesta.

10. Definir la visión general de UML?

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software.

11. Describir los mecanismos comunes y técnicas comunes del modelado de UML?

Los mecanismos comunes son:

- **Especificaciones:** Es más que un lenguaje gráfico (Semántica detrás de la notación)
- **Adornos:** Detalles sobre una clase, nivel de acceso de sus métodos, notas.
- **Divisiones Comunes:** Clase/Objeto o Interfaz/Implementación.
- **Extensibilidad:** Estereotipos, valores etiquetados o restricciones.

12. ¿Qué es un sistema software?

Se conoce como software al sistema formal de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

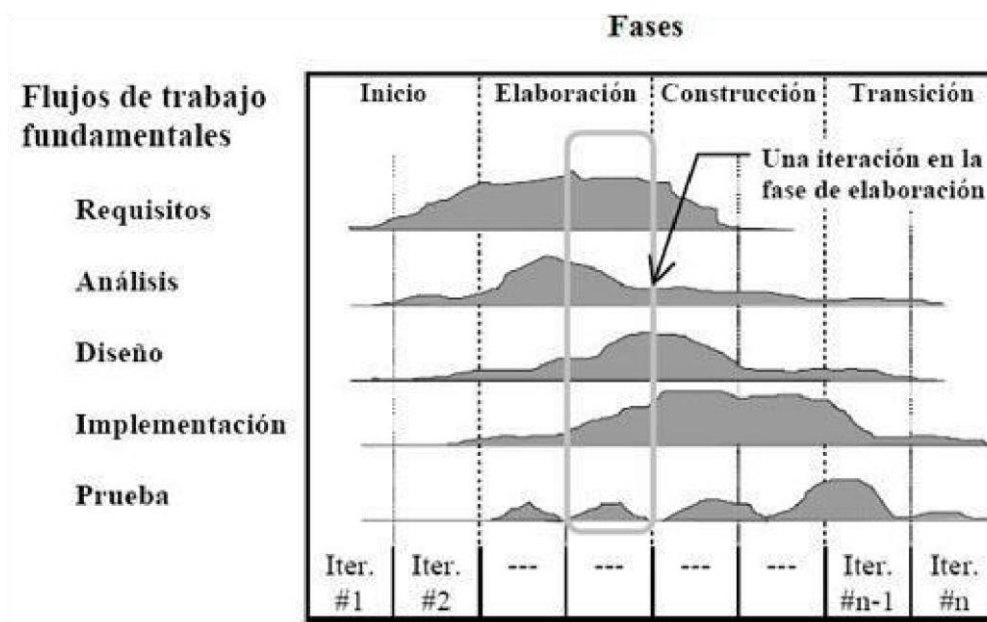
13. ¿Qué es un esbozo y que es un artefacto?

Un esbozo es un artículo muy corto o sin acabar. El esbozo típico no proporciona toda la información necesaria. Sin embargo, eso no quiere decir que el esbozo no sea un artículo legítimo, sólo falta completarlo.

Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software. Otros se enfocan en el proceso de desarrollo en sí mismo, como planes de proyecto, casos de negocios o enfoque de riesgos.

14. ¿Definir y diseñar la vida del proceso unificado y cuáles son las fases dentro de un ciclo?

Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema. Cada ciclo constas de cuatro fases: inicio, elaboración, construcción, y transición.



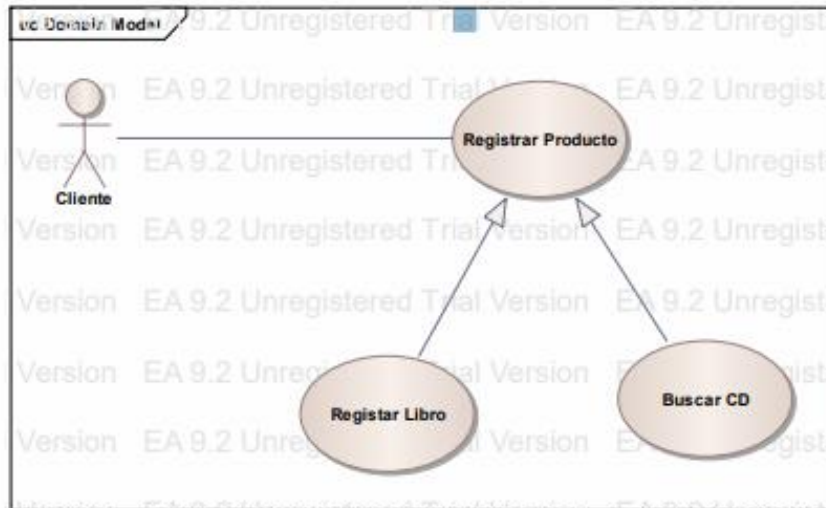
15.Realizar las actividades y en que flujo de trabajo participar los trabajadores del PUDS?

Los trabajadores realizan actividades. Una actividad es algo que realiza un trabajador para proveer un resultado de valor en el contexto de un proyecto.

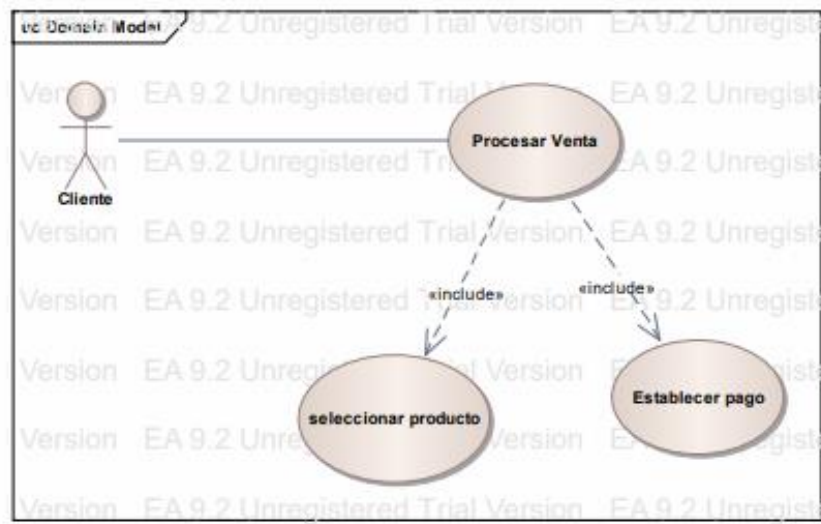
Actividad	Trabajador	Resp. de artefacto (Salida)	Artefactos (entrada)
REQUISITOS			
1 <u>Encontrar actores y casos de uso</u>	Analista de Sistemas	Modelo de casos de uso (esbozado) Glosario	Modelo del negocio o Modelo del dominio Requisitos adicionales Lista de características
2 <u>Priorizar casos de uso</u>	Arquitecto	Descripción de la arquitectura (vista del mod.de casos de uso)	Modelo de casos de uso (esbozado) Requisitos adicionales Glosario
3 <u>Detallar Casos de Uso</u>	Especificador de casos de uso	Caso de uso (detallado)	Modelo de casos de uso (esbozado) Requisitos adicionales Glosario
4 <u>Estructurar el modelo de casos de uso</u>	Analista de Sistemas	Modelo de casos de uso (estructurado)	Modelo de casos de uso (esbozado) Requisitos adicionales Caso de uso (detallado) Glosario
5 <u>Prototipar interfaz de usuario</u>	Diseñador de Interfaz de usuario	Prototipo de interfaz de usuario	Modelo de casos de uso Requisitos adicionales Caso de uso (detallado) Glosario

16. Hacer dos ejemplos de cada una de las relaciones posibles en un diagrama de casos de uso.

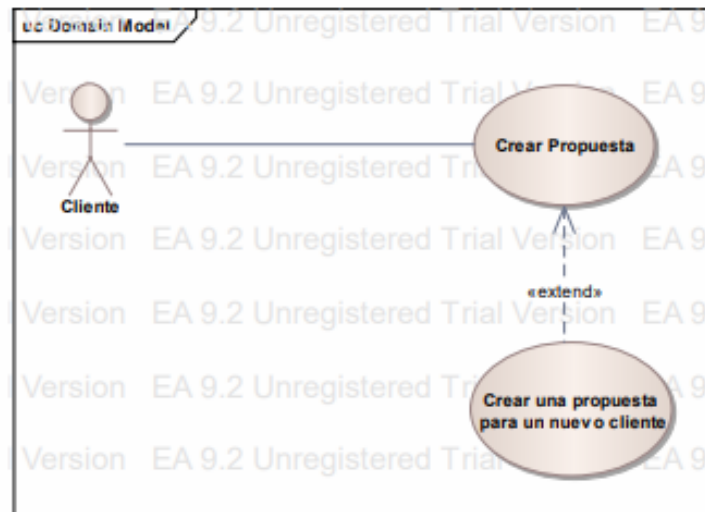
Relación de generalización



Relación inclusión



Relación extended



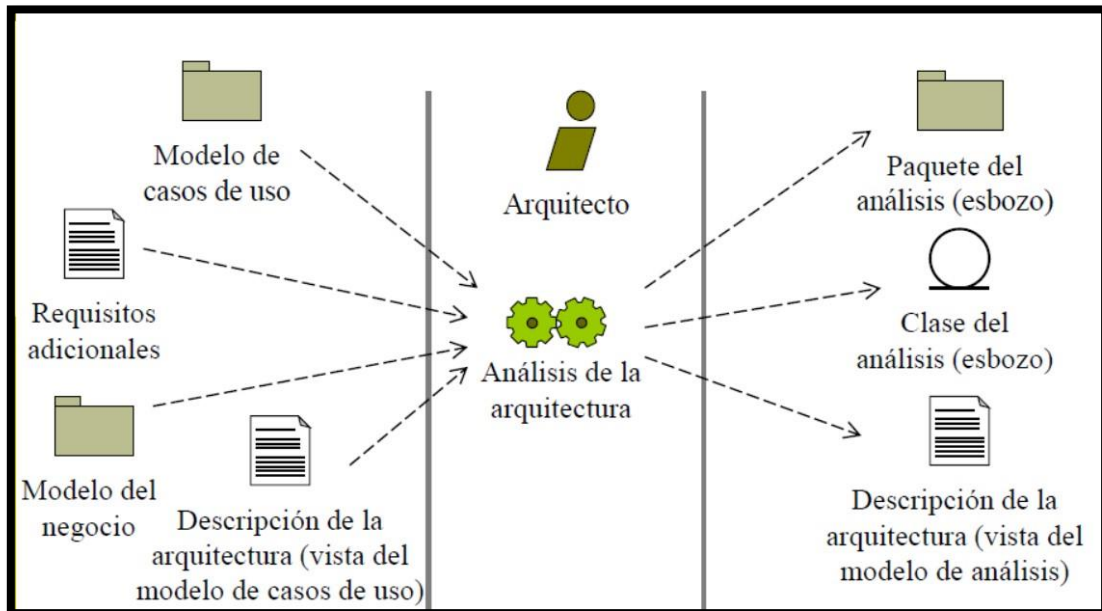
17. Cuando y como se desarrollar un modelo de negocio y un modelo de dominio

El modelado del negocio es una técnica para comprender los procesos de negocio de la organización. Un modelo de objetos del negocio describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y de unidades de trabajo. Cada realización de un caso de uso del negocio puede mostrarse en diagramas de interacción y diagramas de actividad.

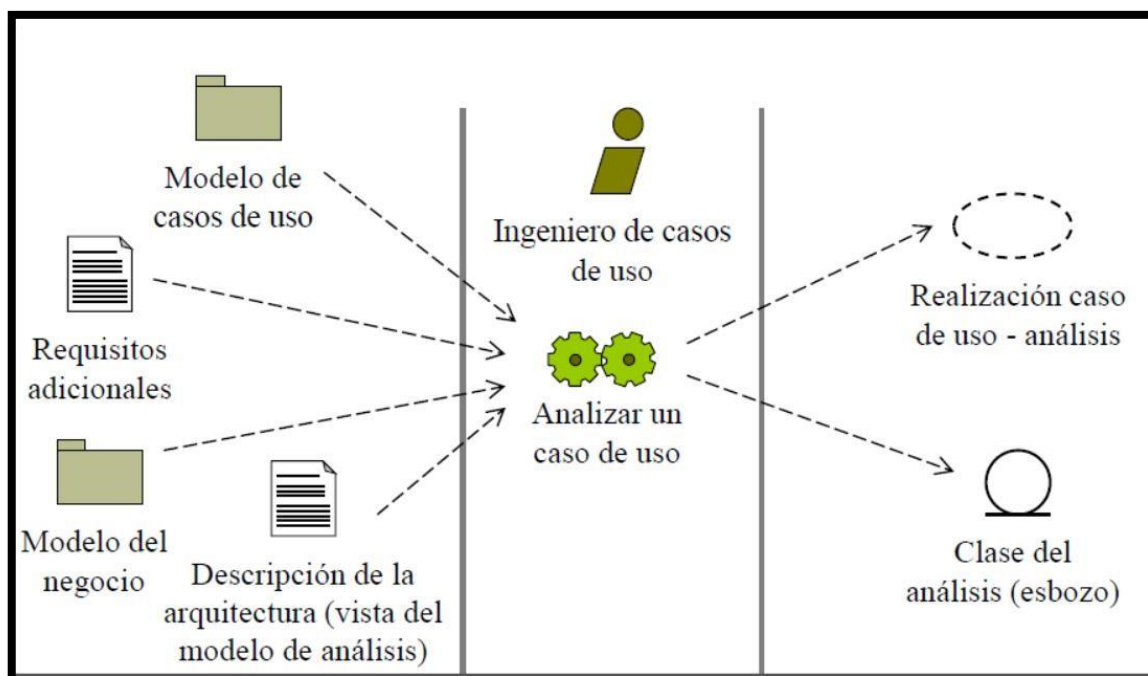
Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de dominio se representa fundamentalmente por diagramas de clases en UML. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

18. Realizar el flujo de trabajo Análisis y especificar la entrada y salida de cada actividad

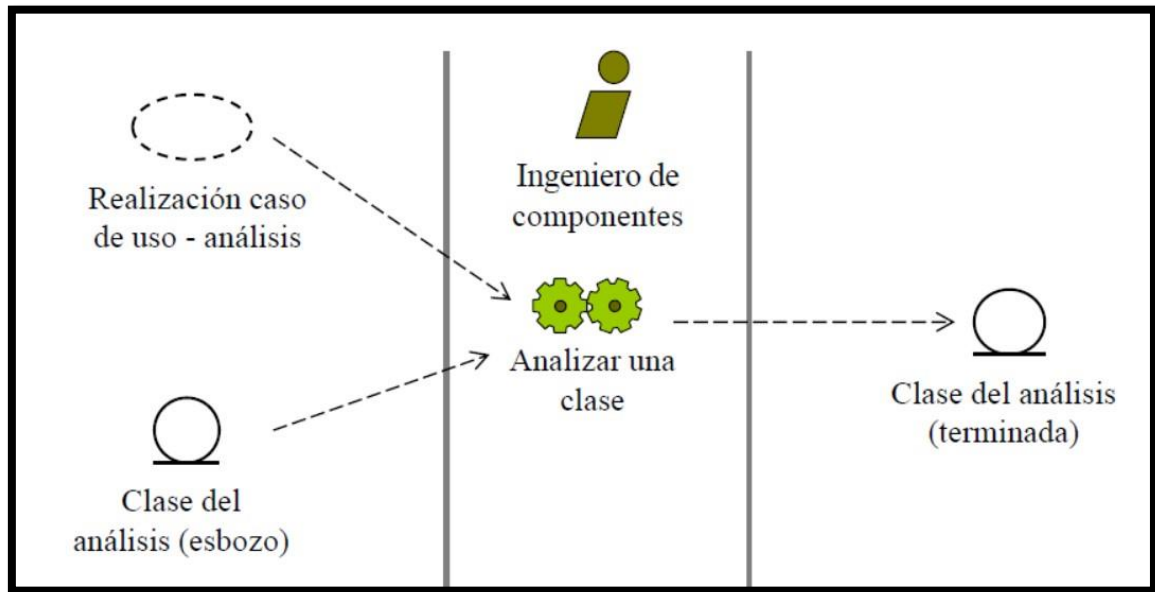
Actividad análisis de la arquitectura



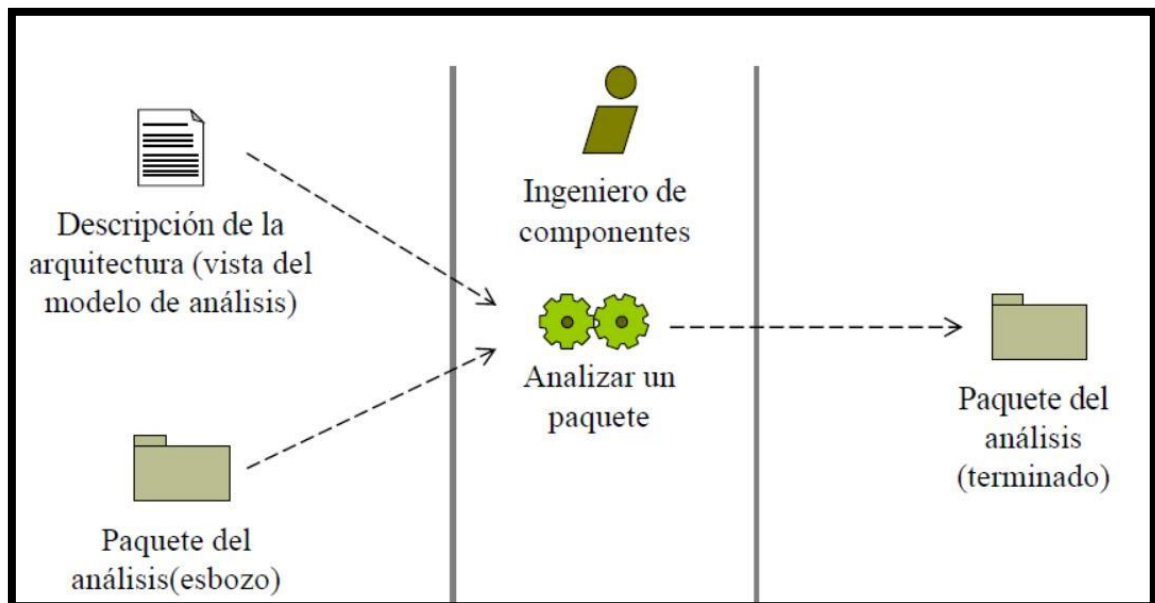
Actividad analizar un caso de uso



Actividad Analizar una clase

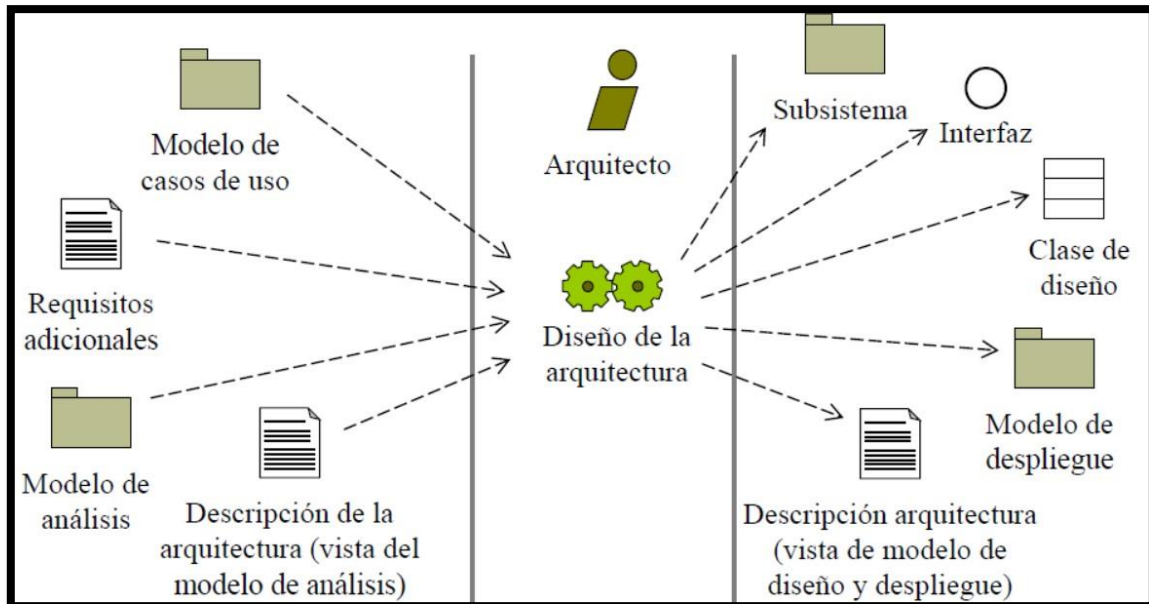


Actividad Analizar un paquete

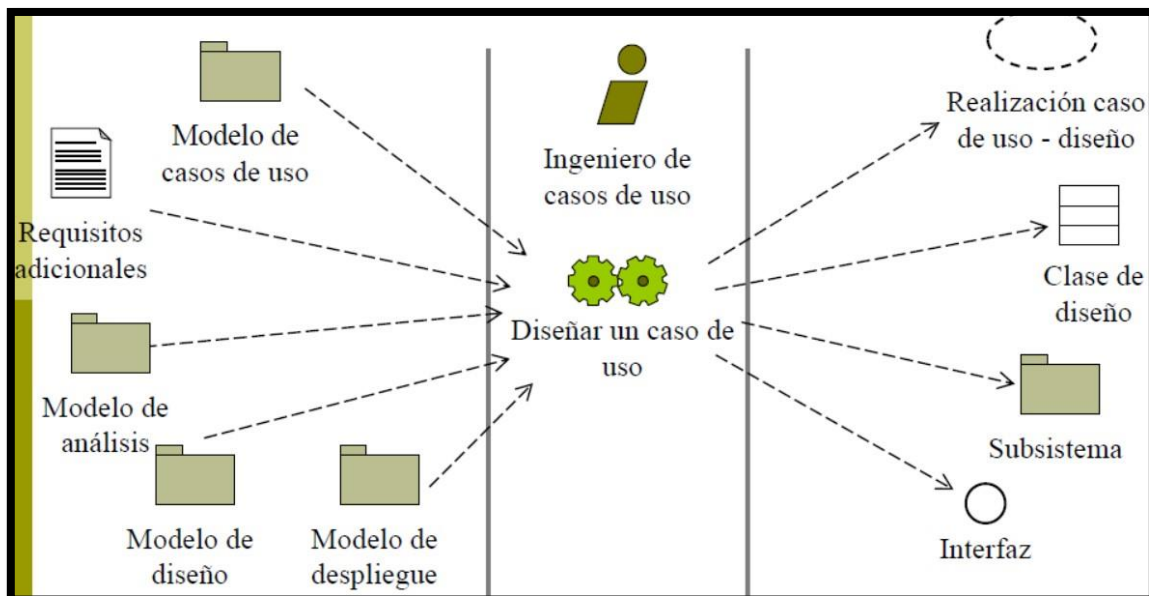


19. Realizar el flujo de trabajo Diseño y especificar la entrada y salida de cada actividad

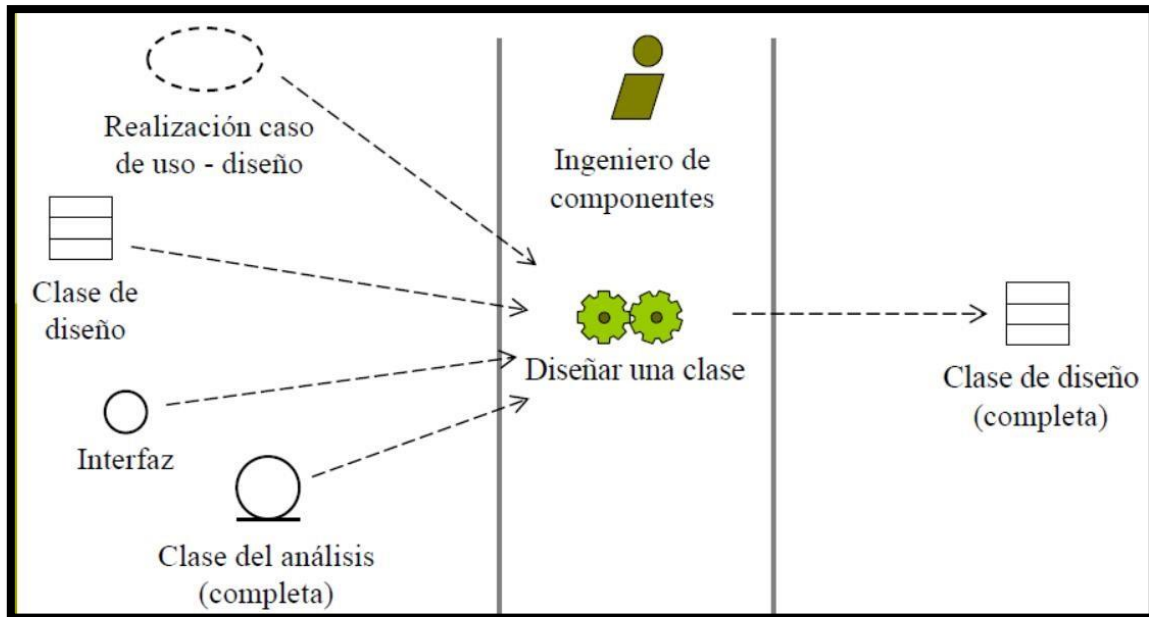
Actividad diseño de la arquitectura



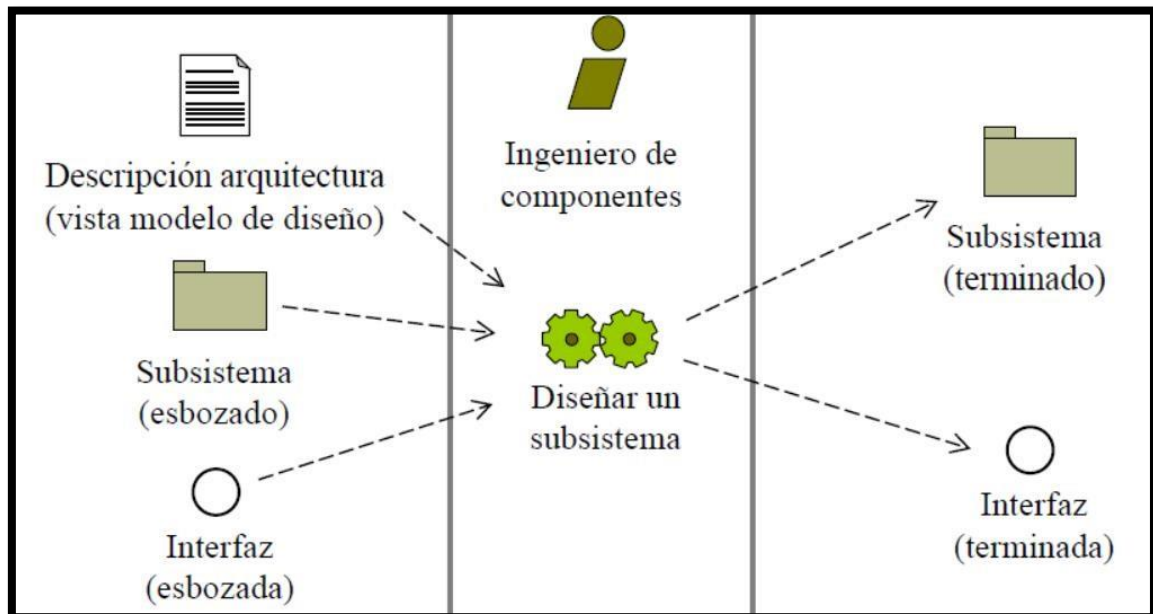
Actividad diseño de un caso de uso



Actividad diseño de una clase



Actividad diseño de un subsistema

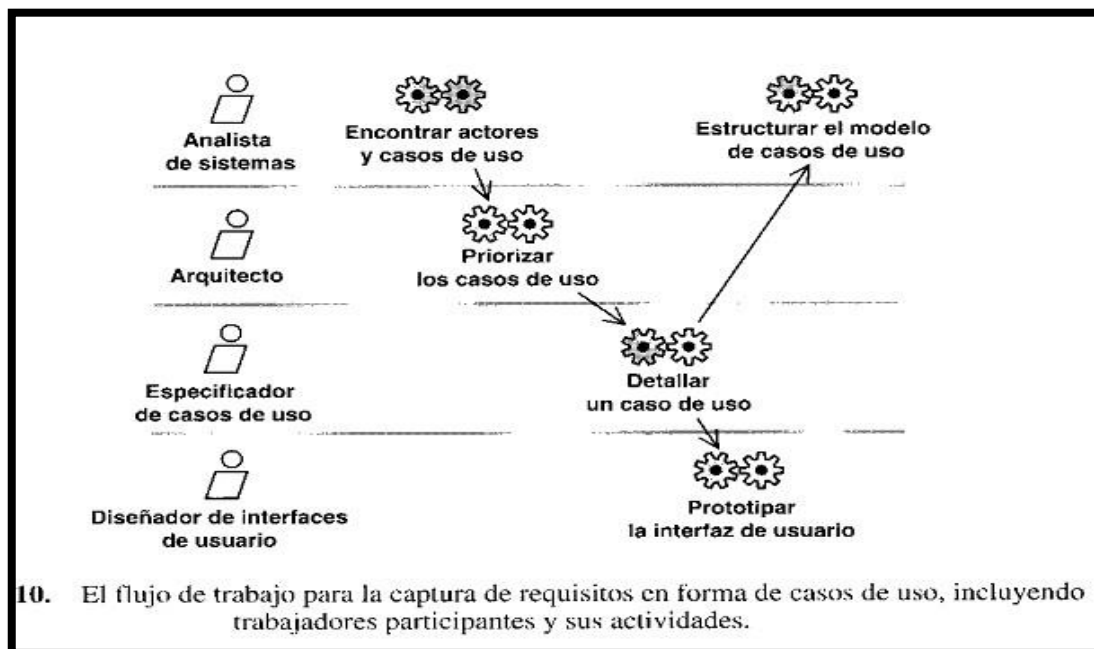


20. En un diagrama de actividad de UML a que se denomina “estado de acción” y “estado de actividad”

El estado de acción se refiere a la acción atómica ininterrumpida ejecutada, y después de completar esta acción, cambia a otro estado al completar la transición.

Un estado de actividad representa una acción, un paso en el flujo de trabajo o en la ejecución de una operación. También describe grupos secuenciales y concurrentes de actividades y actividades enlazadas por transiciones automáticas. Cuando una actividad termina, se desencadena el paso a la siguiente actividad.

21. Usando diagrama de actividades organizado en calles describa el Flujo de Trabajo de captura de requisitos según el PUDS



22. Realizar un diagrama de comunicación para inscripción de estudiantes de la FICCT a un semestre normal

23. Realizar el diagrama de secuencia para inscripción de estudiantes de la FICCT a un semestre normal

24. Explicar las 4 p de PUDS y como lo utilizo en su proyecto?

- **Persona:** Por todos los medios posibles se debe atraer el personal talentoso e inteligente que desea superarse y, sobre todo, desea trabajar en equipo para la realización del proyecto en que participe.

En nuestro caso, las personas elegidas en nuestro proyecto fueron seleccionadas debido a que son personas que nos conocemos en persona y tenemos el conocimiento de que íbamos a trabajar.

- **Producto:** Se denomina productos a todos aquellos artefactos que se creen durante la vida del proyecto, modelos, códigos, ejecutable, documentación, diagramas UML, bocetos de la interfaz de usuario, prototipos, componentes, planes de prueba, ingeniería y gestión, colección de modelos, modelos de casos de uso, análisis, diseño, despliegue, implementación y prueba.

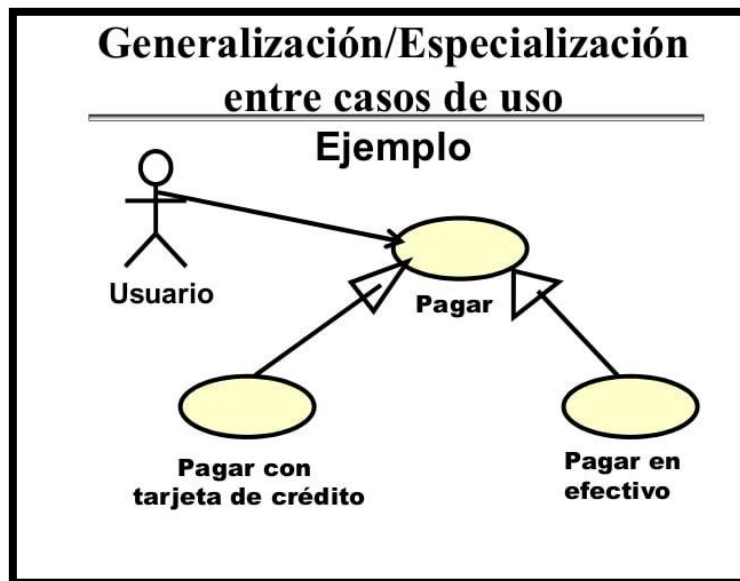
En nuestro proyecto todo el producto se ve reflejado en la página (en el sistema de información), pero los bocetos y el resultado se encuentran realizados y archivados en una computadora y en la documentación del proyecto.

- **Proceso:** Se denomina proceso al conjunto de actividades que se realizan para crear el producto (Plantilla para crear el proyecto). El proceso se define en términos de flujos de trabajo (conjunto de actividades), se identifican trabajadores y artefactos, además de que se utilizan diagramas de actividad de UML para describir los flujos de trabajo.

- **Proyecto:** Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Es un elemento organizativo de gestión que establece una secuencia de cambios, por el cual va evolucionando diariamente.

En nuestro caso, tratamos todo lo posible para reparar errores y aumentar más factores a nuestro sistema de información.

25. En un diagrama de casos uso muestre el uso de generalización / especialización y también desarrollar cuando utilizar actores que visualicen herencia.



26. Definir y especificar los artefactos de SCRUM?

Los artefactos son todos los elementos que te garantizan la transparencia y el registro de la información fundamental del proceso de SCRUM. Dicho de otra manera, son los recursos que cimientan la productividad y la calidad de cualquier proyecto.

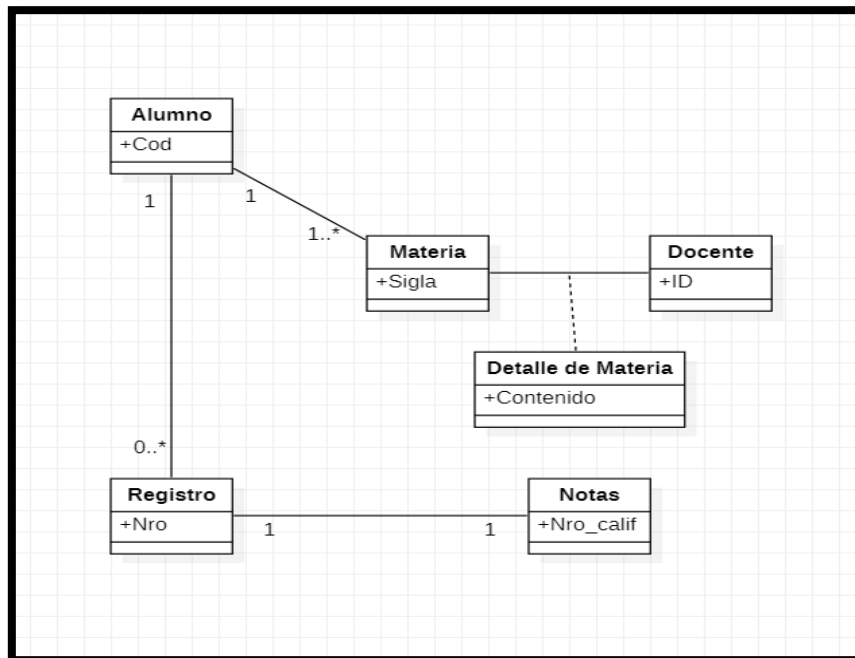
Los artefactos son:

Product Backlog: El product Backlog (PB) es una lista ordenada con todo lo que necesita un producto para cumplir las necesidades de los clientes potenciales y única fuente de requisitos para realizar modificaciones en él.

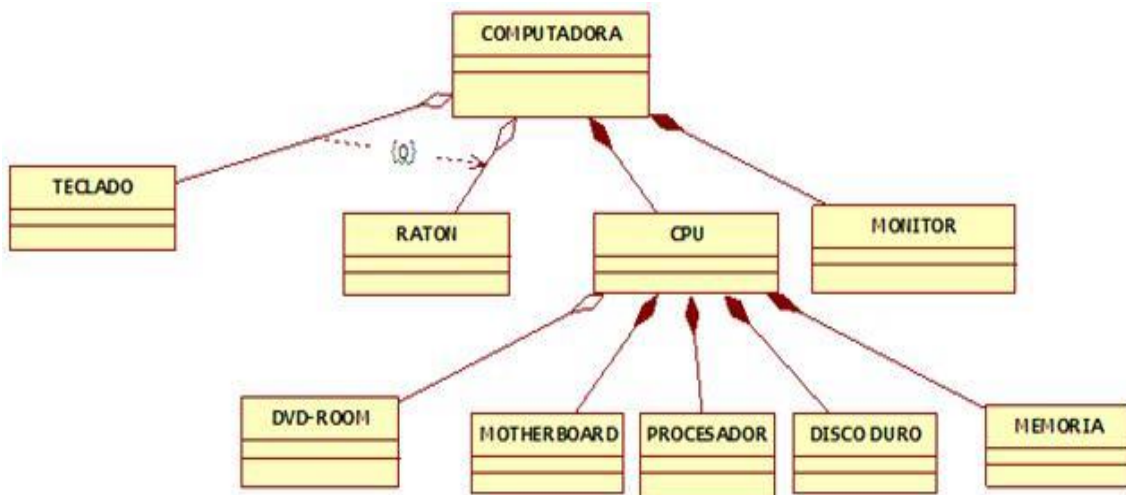
Sprint Backlog: Es un subconjunto de elementos del PB elegidos para abordarse en el periodo de tiempo o sprint más un plan para ofrecerlos como incremento del producto y lograr el objetivo del sprint.

Incremento: Se trata del resultado del sprint. Un entregable utilizable y potencialmente desplegable.

27. Haga un diagrama de clases para el registro de notas de alumnos, considere que un alumno lleva más de una materia, y que una misma materia puede ser dictada por más de un docente



28. Realice ejemplos utilizando herencia, agregación y composición entre clases



29. Describir las características del análisis de la arquitectura y análisis de paquete ?

Las características de la arquitectura son:

Automatización

La automatización elimina la fricción que hacía que los sistemas de datos heredados fueran tediosos de configurar..

Seguridad

La seguridad está integrada en la arquitectura de datos moderna, lo que garantiza que los datos estén disponibles en función de la necesidad de conocimiento definida por la empresa.

Orientación al usuario

En el pasado, los datos eran estáticos y el acceso era limitado. Los responsables de la toma de decisiones no obtenían necesariamente lo que querían o necesitaban, sino lo que estaba disponible.

Las características de análisis de paquete son:

- El número de análisis que realizan y profundidad de los mismos.
- El nivel de conocimiento requerido por el analista.
- El volumen de datos que son capaces de tratar.
- La ergonomía de uso.
- La facilidad de programación

Los paquetes del análisis proporcionan un medio de organizar los artefactos del modelo de análisis en piezas manejables. Un paquete de análisis puede constar de clases de análisis, de realización de casos de uso, y de otros paquetes de análisis (recursivamente).

Deben ser cohesivos y débilmente acoplados

Tienen las siguientes características:

- Pueden representar una separación de intereses de análisis

- Han de crearse basándose en los requisitos funcionales y en el dominio del problema
- Probablemente se convertirán en subsistemas

30. Concepto de diagrama de componente y explicar su contexto?

Un diagrama de componentes es un diagrama tipo del Lenguaje Unificado de Modelado.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Debido a que los diagramas de componentes son más parecidos a los diagramas de casos de usos, estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

31. Bajo qué criterios decide usar los estereotipos <<include>> y <<extended>> en un diagrama de casos de uso

<<extended>> se usa cuando un caso de uso agrega pasos a otro caso de uso de primera clase.

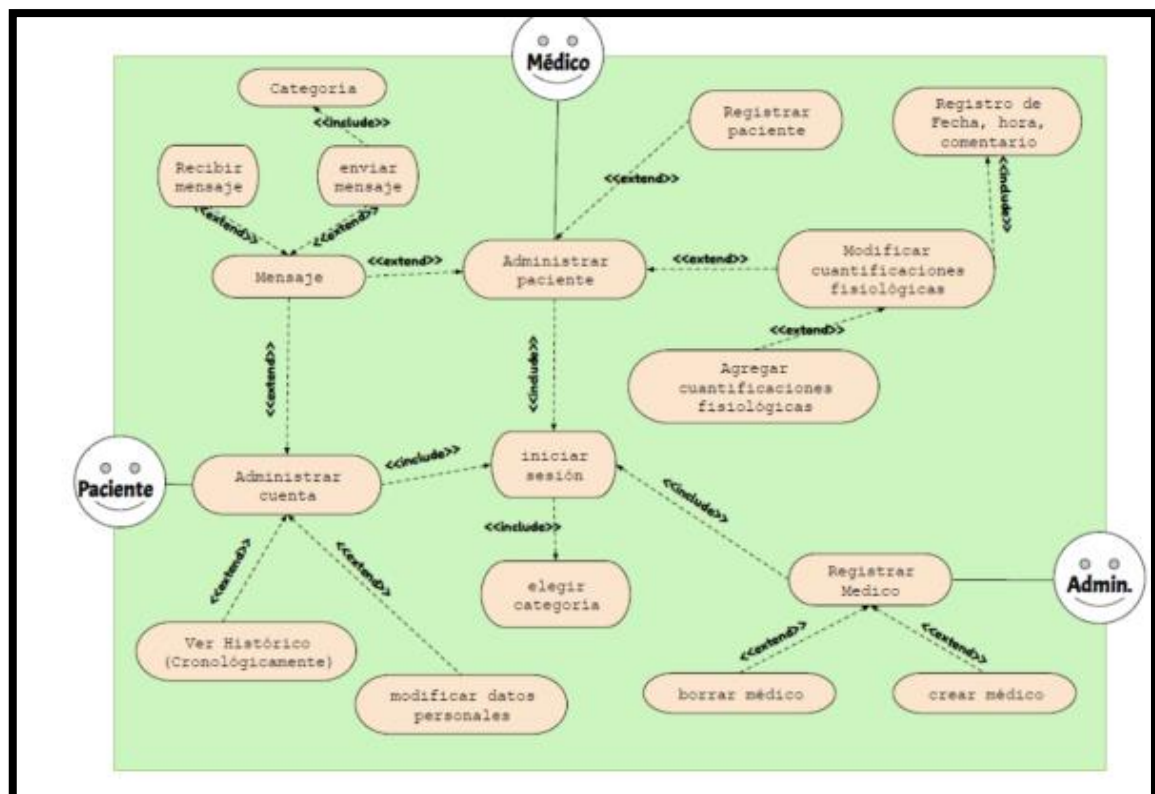
Por ejemplo, imagine que "Retirar efectivo" es un caso de uso de un cajero automático (ATM). La "Tarifa de evaluación" extendería el Retiro en efectivo y describiría el condicional "punto de extensión" que se crea cuando el usuario del cajero automático no realiza operaciones bancarias en la institución propietaria del cajero automático. Observe que el caso de uso básico de "Retirar efectivo" se mantiene por sí solo, sin la extensión.

<<include>> se usa para extraer fragmentos de casos de uso que son duplicados en casos de uso múltiples. El caso de uso incluido no puede ser independiente y el caso de uso original no está completo sin el incluido. Debe usarse con moderación y solo en los casos en que la duplicación sea significativa y exista por diseño (en lugar de por coincidencia).

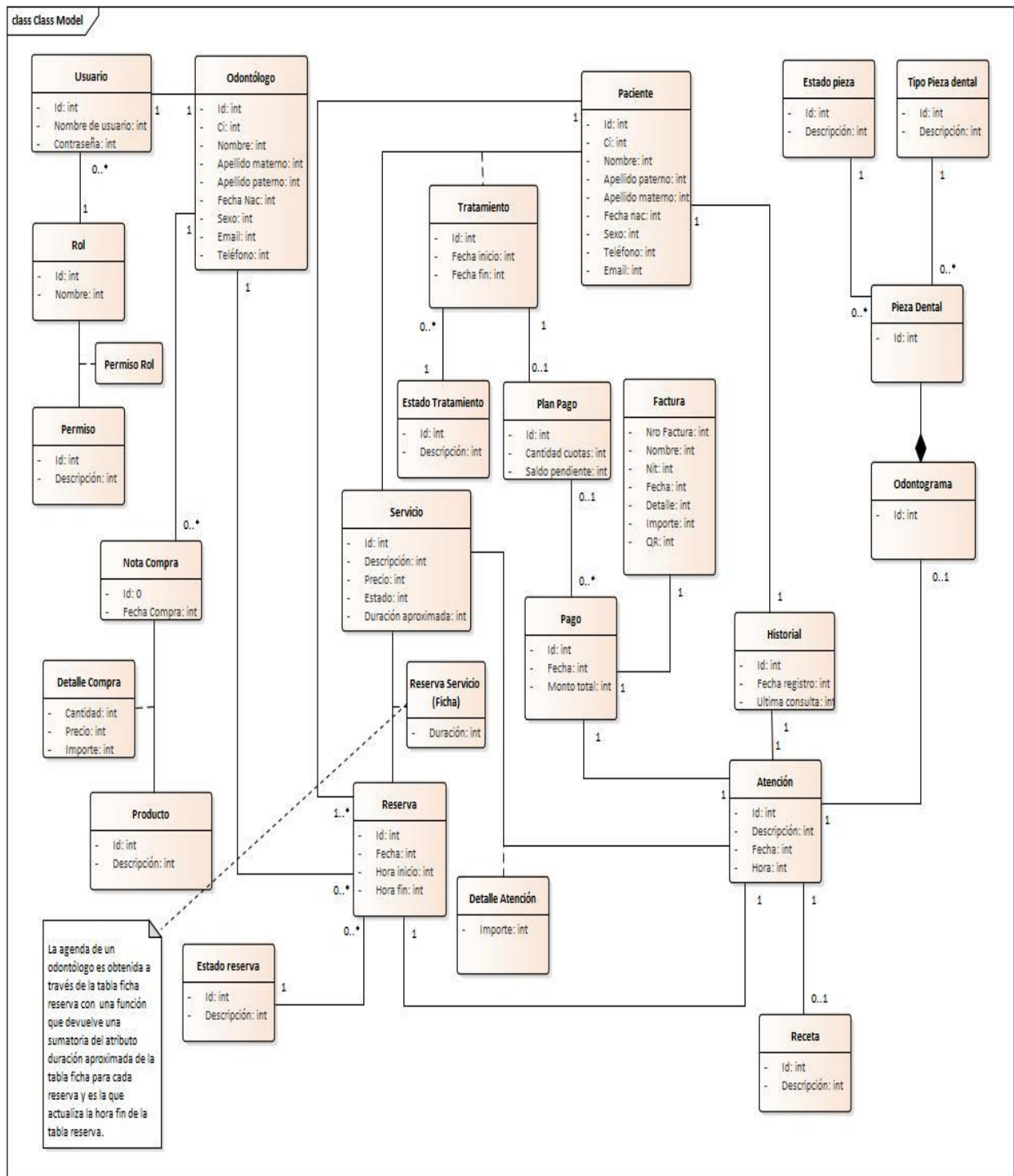
Por ejemplo, el flujo de eventos que se produce al comienzo de cada caso de uso de ATM (cuando el usuario coloca su tarjeta de cajero automático, ingresa su PIN y aparece el menú principal) sería un buen candidato para una inclusión.

32. Caso de estudio: Sistema de información para administrar pacientes de una unidad sanitaria que tiene diferentes especialidades de médicos que atienden en horarios definidos y realizan seguimientos y tratamientos por cada paciente.....

a) Realizar el diagrama de casos de uso



c) Analizar casos de uso



c) Analizar casos de uso

d) Análisis de clases