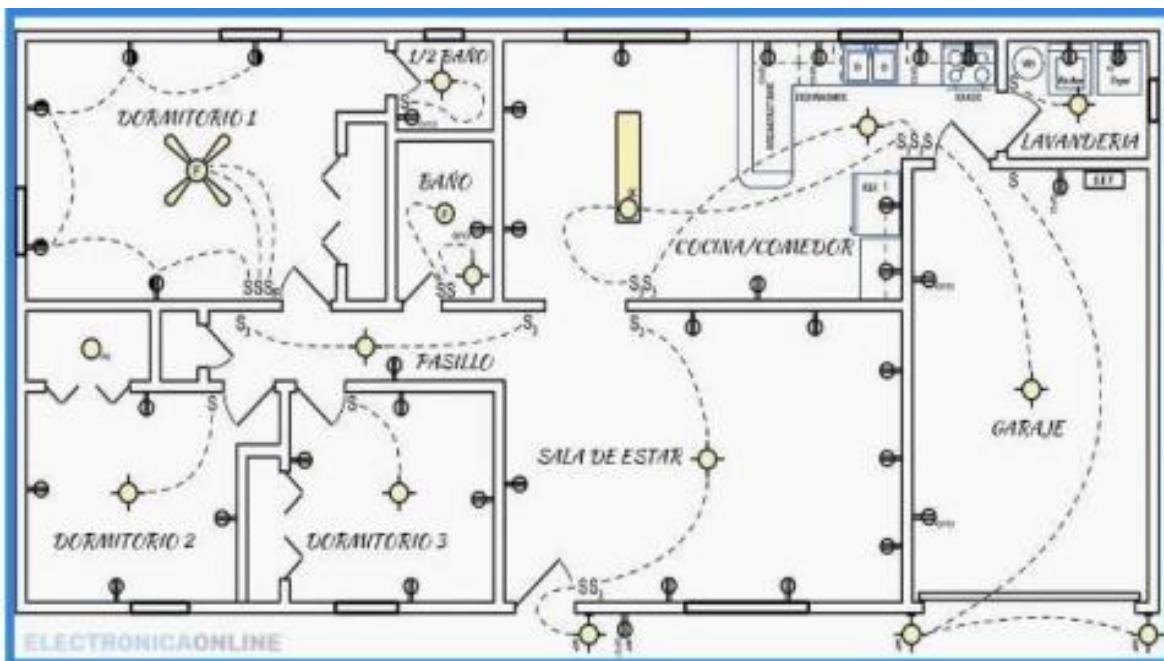


2. Actividad de Contextualización e Identificación de Conocimientos Previos.

Actividad individual.



Describa qué puede ver en el plano.

- Puedo ver que es un plano eléctrico donde se evidencia el circuito el cableado y diversas cosas como interruptores; bombillos y tomas de corriente.

Responda las siguientes preguntas:

¿Cree que, con sus conocimientos nulos de electricidad, pudo identificar de qué diagrama se trataba?

- Se trataba de un plano eléctrico

¿Cree entender dónde están los interruptores?

- Los interruptores se representan con la letra (S)

¿Cree saber dónde están los bombillos?

- Hay diferentes tipos de bombillos en ese plano, puedo evidenciar bombillos de pared de techo y empotradas en el techo, tienen un color amarrillo que hace referencia a la luz.

¿Cuántos dormitorios hay?

- En total hay 3 dormitorios

3.3. Actividad de Apropiación del Conocimiento (Conceptualización y Teorización).

En el proceso del desarrollo de software, hay que generar diagramas que permitan entender la estructura y el comportamiento del software, por lo tanto, se deben desarrollar Diagramas para la especificación y análisis de requisitos levantados en la primera parte de la formación.

Tema: “Lenguaje Unificado de Modelado (UML).” Responda las siguientes preguntas:

Pregunta 1: ¿Cuál es la importancia de UML?

La importancia del lenguaje unificado de modelado (UML) este nos permite documentar, diseñar, modelado y el análisis de un sistema de software, este lenguaje tiene tres características muy importantes que son: visual, este nos permite utilizar diagramas para los diferentes aspectos de un sistema, también es muy versátil y estándar.

Pregunta 2: Beneficios de usar UML en proyectos de desarrollo de software

- Comunicación clara y efectiva: Lenguaje común, visualización de ideas.
- Mejor diseño y análisis del sistema: Diseño modular, identifica posibles problemas, flexibilidad y escalabilidad.
- Documentación estándar y reusable: Referencia futura
- Facilita la colaboración: Equipo o stakeholders.
- Compatible con diferentes metodologías.
- Mejora en la calidad del software: Código organizado, pruebas más efectivas.
- Ahorro de tiempo y costos: Automatización
- Aplicación de diversos dominios

Pregunta 3: ¿Cuáles son los diagramas que pueden representarse con UML?

El lenguaje unificado de modelado (UML) los diagramas se dividen en dos partes, Diagramas estructurales y Diagramas de comportamiento.

DIAGRAMAS ESTRUCTURALES

Estos diagramas muestran la estructura estática del sistema, incluyendo sus componentes, relaciones y organización.

- Diagrama de clases: Representa las clases del sistema, sus atributos, métodos y relaciones (herencia, asociación, composición, etc.).
- Diagrama de objetos: Muestra instancias específicas de las clases, útil para visualizar ejemplos concretos del sistema.
- Diagrama de componentes: Representa los componentes físicos del sistema (por ejemplo, bibliotecas, módulos) y cómo interactúan entre sí.
- Diagrama de despliegue: Modela la arquitectura física del sistema, mostrando cómo se distribuyen los componentes en nodos (servidores, dispositivos, etc.).
- Diagrama de paquetes: Agrupa elementos del modelo en paquetes para organizar y estructurar el diseño.
- Diagrama de perfil: Permite extender UML mediante estereotipos y etiquetas personalizados, adaptándolo a necesidades específicas.

DIAGRAMAS DE COMPORTAMIENTO

Estos diagramas muestran cómo el sistema responde a eventos, cómo interactúan sus elementos y cómo evolucionan en el tiempo.

- Diagrama de casos de uso: Muestra las funcionalidades del sistema desde la perspectiva del usuario o de los actores involucrados.
- Diagrama de secuencia: Representa cómo los objetos del sistema interactúan entre sí en orden temporal para cumplir un objetivo.
- Diagrama de comunicación: Similar al diagrama de secuencia, pero se enfoca en la relación entre los objetos en lugar del orden de los mensajes.
- Diagrama de actividades: Modela flujos de trabajo o procesos, mostrando actividades, decisiones y transiciones.
- Diagrama de estados (o de estado-transición): Representa los diferentes estados de un objeto a lo largo de su ciclo de vida y los eventos que causan cambios entre estados.
- Diagrama de interacción general: Proporciona una vista de alto nivel de las interacciones entre los componentes del sistema.
- Diagrama de sincronización (timing diagram): Representa cómo cambia el estado de los objetos en función del tiempo, útil en sistemas sensibles al tiempo.

Pregunta 4: Diferencias entre diagramas estructurales, de comportamiento e interacción.

La principal diferencia es que los diagramas estructurales se encargan de los componentes que tiene el sistema y los diagramas de comportamientos se encargan de como interactúan y se comportan los componentes y responder algún evento posible.

Pregunta 5: Ejemplos de casos en los que UML es utilizado en la industria.

Desarrollo de Software

- Diagrama de casos de uso: Define las funcionalidades como "Registrar cliente", "Actualizar datos" o "Generar reporte de ventas".
- Diagrama de clases: Modela entidades como Cliente, Pedido, Producto, con sus atributos y relaciones.
- Diagrama de secuencia: Representa la interacción entre el usuario, el sistema y la base de datos al registrar un cliente.

Telecomunicaciones

- Diagrama de componentes: Modela los módulos como "Facturación", "Gestión de clientes" y "Cobro".
- Diagrama de secuencia: Muestra el flujo de datos desde la generación de una factura hasta su cobro y actualización en el sistema.
- Diagrama de sincronización: Describe cómo los diferentes servicios (voz, datos) interactúan en tiempo real con los sistemas de cobro.

Salud

- Diagrama de casos de uso: Define funcionalidades como "Registrar paciente", "Agendar consulta" y "Acceder al historial médico".
- Diagrama de actividades: Modela el flujo de atención desde la admisión del paciente hasta su alta.
- Diagrama de clases: Representa entidades como Paciente, Médico, Consulta, con sus atributos y relaciones.

Pregunta 6: ¿Qué herramientas de software existen en el mercado para UML?

1. Enterprise Architect
2. Vicual paradigm
3. IBM Rational Rose (Rational Software Architect)
4. Astah Professional
5. StarUML