



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es)**

Durante esta actividad los aprendices deben estar prestos a la contextualización del tema que brinda el instructor. Las actividades se realizan de forma presencial en la formación con la orientación del instructor y de manera autónoma con la revisión del material de apoyo, referencias bibliográficas y con la elaboración de actividades.

- **Ambiente Requerido**

El ambiente de aprendizaje debe estar conformado por:

22 equipos con los requerimientos mínimos:

- Sistema operativo: Windows
- Disco Duro: 400 GB
- Ram: 2GB como mínimo
- Procesador: Intel Core 2Duo de 2,66 Mhz
- Mesas y sillas

1 servidor de aplicaciones

Conexión a Internet permanente

- **Materiales**

Computadores de escritorio y portátiles con acceso a internet, software de aplicación para realizar informes y visualizar material digital.

3.1. Actividades de Reflexión Inicial

El UML (Lenguaje Unificado de Modelado), es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que los creadores de sistemas generan diseños que capturen sus ideas en forma convencional y fácil de aprender para comunicarlas a otras personas. El UML es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson. “Los tres Amigos” como se apodaron estos tres grandes utilizaban metodologías diferentes para el diseño de software hasta que en los años 90 decidieron unirse y trabajar en conjunto en una sola metodología, el UML está compuesto de diversos elementos gráficos que se combinan para conformar diagramas, con las mismas finalidades que es presentar diversas perspectivas de un sistema a las cuales se conoce como modelo.

Fuente https://www.ecured.cu/Diagrama_de_Clase

El diagrama de clases hace parte de los diagramas UML y muestra la estructura estática del sistema que se está modelando.

¿Cada aprendiz debe analizar cuáles son las características esenciales de los objetos, el ser humano hace parte de los objetos para un sistema? Justifique su respuesta.

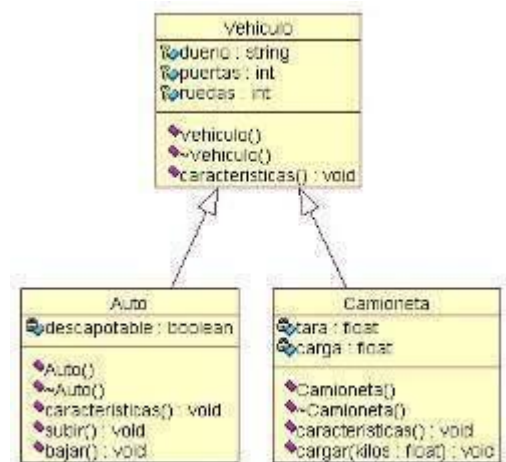


3.2. Actividades de Contextualización e Identificación de Conocimientos Necesarios para el Aprendizaje

3.2.1. En parejas de trabajo los aprendices deben investigar para realizar una presentación que responda a los siguientes interrogantes:

1. ¿Qué es una clase?
2. ¿Qué elementos hacen parte de la clase, describa cada uno?

Seguidamente el instructor realiza una contextualización a los aprendices acerca de los interrogantes propuestos.



La Programación Orientada a Objetos supone un cambio en la concepción del mundo de desarrollo de software, introduciendo un mayor nivel de abstracción que permite mejorar las características del código final. De manera muy básica, las aportaciones de este paradigma se pueden resumir en:

- Conceptos de clase y objeto, que proporcionan una abstracción del mundo centrada en los seres y no en los verbos.
- Los datos aparecen encapsulados dentro del concepto de clase. El acceso a los datos se produce de manera controlada e independiente de la representación final de los mismos. Como consecuencia, se facilita el mantenimiento y la evolución de los sistemas, al desaparecer las dependencias entre distintas partes del sistema.
- Mediante conceptos como la composición, herencia y polimorfismo se consigue simplificar el desarrollo de sistemas. La composición y la herencia nos permiten construir clases a partir de otras clases, aumentando en gran medida la reutilización.

3.2.2. Investigue, analice y conceptualice acerca de UML, consultando los siguientes temas:

- a) ¿Qué se entiende por programación orientada a objetos?
- b) ¿Cuál es la diferencia entre programación estructurada y programación orientada a objetos?
- c) ¿Qué es una clase y cuáles son los elementos que la componen?
- d) ¿Qué es un objeto?
- e) ¿Qué es herencia, de un ejemplo?
- f) ¿Qué se entiende por encapsulación?
- g) ¿Qué es polimorfismo?
- h) ¿Explique los diferentes tipos de relaciones de un diagrama de clases UML: asociación, agregación, dependencia, composición...; con que línea se representa cada una?
- i) ¿Qué es cardinalidad de las relaciones y como se clasifican?
- j) ¿Qué es un diagrama de clases?



3.3. Actividades de Apropiación

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de agregación, ya que una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica; mostrando un conjunto de elementos que son estáticos, como las clases y tipos junto con sus contenidos y relaciones. Un diagrama de clases está compuesto por los siguientes elementos: Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso. Fuente https://www.ecured.cu/Diagrama_de_Clase

Las clases se representan por rectángulos que muestran el nombre de la clase y opcionalmente el nombre de las operaciones y atributos. Los compartimientos se usan para dividir el nombre de la clase, atributos y operaciones. Adicionalmente las restricciones, valores iniciales y parámetros se pueden asignar a clases.

La notación que precede el nombre del atributo u operación indica la visibilidad del elemento, si se usa el símbolo + el atributo y la operación tienen un nivel público de visibilidad, si se usa un símbolo – el atributo u operación es privado. Además, el símbolo # permite definir una operación o atributo como protegido y el símbolo ~ indica la visibilidad del paquete.

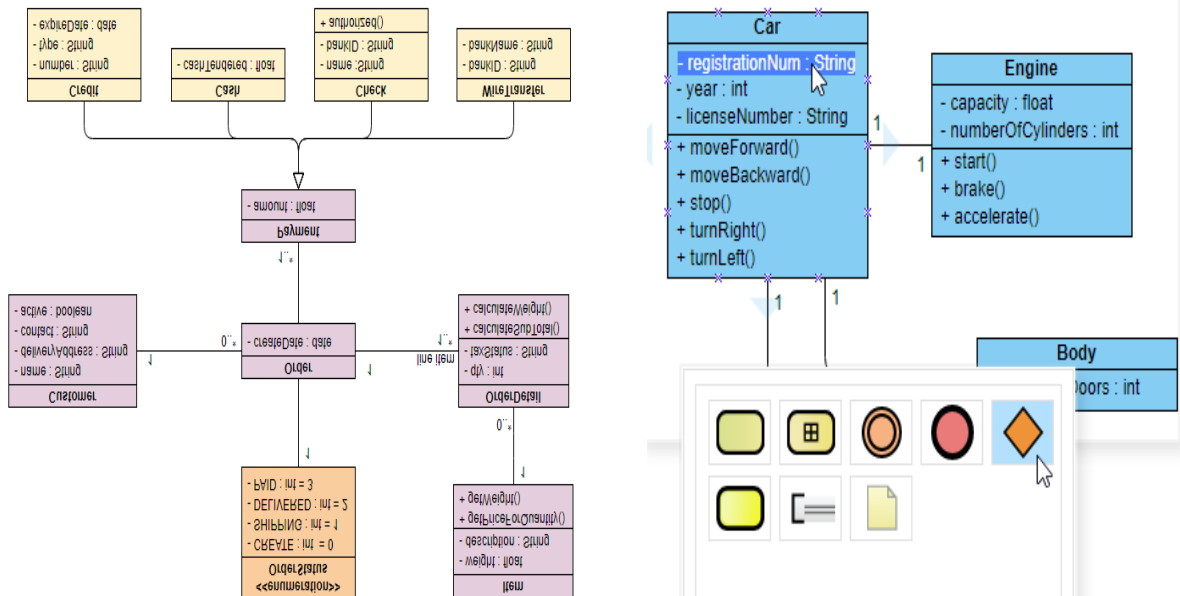
Fuente https://www.ecured.cu/Diagrama_de_Clase

En parejas de trabajo explora la siguiente terminología de acuerdo con las instrucciones del instructor: ¿Qué significa instancia, atributo, método, visibilidad, herencia, asociación, agregación, composición, multiplicidad en diagramas de clase?

- Investigue tres ejemplos y detalle su aplicación

3.4. Actividades de Transferencia del Conocimiento

Visual Paradigm Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Esta herramienta tiene disponibilidad en múltiples plataformas (Windows, Linux), Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación, Modelo y código que permanece sincronizado en todo el ciclo de desarrollo, Disponibilidad de múltiples versiones, con diferentes especificaciones.



Fuente https://www.ecured.cu/Visual_Paradigm

La anterior es una de las múltiples herramientas de software que encontramos para construir diagramas de clase, como actividad en parejas de acuerdo a las indicaciones del instructor desarrollar los diagramas propuestos para los casos de estudio y en los equipos de trabajo del **proyecto formativo**, de acuerdo a los casos uso planteados, modelo relacional notación crow's foot construir los diagramas de clase pertinentes.

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: Prueba en línea sobre conceptos fundamentales de diagramas UML clases.	Identifica las principales características para la construcción del diagrama de clases.	Lista chequeo Cuestionario en línea GFPI-F-135 V01



<p>Evidencias de Desempeño</p> <p>Desarrollo de Talleres y actividades en la formación.</p> <p>Sustentación de los diagramas elaborados.</p> <p>Valoración actitudinal frente al proceso.</p> <p>Evidencias de Producto:</p> <p>Desarrollo del diagrama de clases del proyecto formativo.</p> <p>Taller diagrama de clases</p>	<p>Participa activamente con las actividades propuestas en los talleres.</p> <p>Entrega las actividades con pertinencia, vigencia, autenticidad y calidad.</p> <p>Elabora los diagramas de clases dando uso a los elementos y las relaciones que le corresponde, utilizando de manera correcta los recursos de la herramienta.</p>	<p>Observación directa</p> <p>Lista de chequeo</p> <p>Cuestionario</p> <p>Lista de chequeo</p>
--	--	--

5. GLOSARIO DE TÉRMINOS

Diagrama de Clase: El diagrama de clase representa clases, sus partes y la forma en la que las clases de los objetos están relacionados con otro.

Clases: son descripciones de un juego de objetos con características, comportamiento, relaciones y semánticas comunes. Se usan para modelar un juego de conceptos o entidades.

Atributos: Son descripciones de características, se usan para modelar información asociada con una entidad, es un dato importante que contiene valores de esa clase (campos-variables-propiedades).

Métodos Operaciones o Funciones: Son descripciones del comportamiento, se usan para modelar los servicios u operaciones asociados con una entidad, esto es, lo que una entidad puede hacer.

6. REFERENTES BIBLIOGRÁFICOS

Senn, James A.. Análisis y diseño de sistemas de información, McGraw-Hill Interamericana, 1992. ProQuest Ebook Central, <https://ebookcentral-proquest-com.bdigital.sena.edu.co/lib/senavirtualsp/detail.action?docID=3194990>.

Kendall, Kenneth y Julie. Análisis y Diseño de Sistemas. Pearson Educación. 6ª edición. México 2005.

Kendall, K. E., & Kendall, J. E. (2005). Análisis y Diseño de Sistemas (Sexta ed.). México: Pearson Educación, GFPI-F-135 V01

• Sommerville, I. (2005). Ingeniería del Software (Septima ed.). Madrid: Pearson Educación.



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Paola Tatiana Tovar Yaqueline Chavarro	Instructor	Centro de Electricidad, Electrónica y Telecomunicaciones	Julio 2019

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Javier Yara	instructor	CEET	Febrero 2023	Cambio de formato y actualización de contenido