ACTIVIDAD 2

ALEJANDRO STYVENS MONTOYA GAVIRIA

UNIVERSIADAD CATOLICA DEL NORTE

LENGUAJES DE PROGRAMACIONCIUDAD

2021

ACTIVIDAD 2 (PROGRAMACIÓN ORIENTADA A OBJETOS, TEMAS CLASES,

OBJETOS, CONSTRUCTORES Y HERENCIA)

ALEJANDRO STYVENS MONTOYA GAVIRIA

REPASAR Y AFIANZAR LOS TEMAS VISTOS SOBRE PROGRAMACIÓN ORIENTADA A OBJETOS EN

JAVA.

ALEXANDER NARVAEZ BERRIO

UNIVERSIADAD CATOLICA DEL NORTE

LENGUAJES DE PROGRAMACIONCIUDAD

2021

ACTIVIDAD 2 (PROGRAMACIÓN ORIENTADA A OBJETOS, TEMAS CLASES,

OBJETOS, CONSTRUCTORES Y HERENCIA)

AGRADECIMIENTOS

Agradezco a la universidad católica del norte por el compromiso académico y al profesor alexander narvaez berrio por la calidad de las clases que hacen el entendimiento mas ameno.

CONTENIDO

Pág.

[1. INTRODUCCIÓN 8](#_Toc414724022)

[2. OBJETIVOS 9](#_Toc414724023)

[2.1 OBJETIVO GENERAL 9](#_Toc414724024)

[2.2 OBJETIVOS ESPECÍFICOS 9](#_Toc414724025)

[3 PLANTEAMIENTO DEL PROBLEMA 10](#_Toc414724026)

[3.1 DEFINICIÓN DEL PROBLEMA 10](#_Toc414724027)

[3.2 JUSTIFICACIÓN 11](#_Toc414724028)

[5.1 MATERIALES 12](#_Toc414724030)

[5.2 METODOLOGÍA 12](#_Toc414724031)

[6 DESARROLLO DEL PROYECTO 13](#_Toc414724032)

[6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO 13](#_Toc414724033)

[CONCLUSIONES 14](#_Toc414724035)

LISTA DE ANEXOS

Repositorio de GitHub <https://github.com/Alejo0508/actividad-2>

GLOSARIO

Clase: Una clase es una plantilla que define la forma de un objeto; en ella se agrupan datos y métodos que operarán sobre esos datos

Método:Los métodos son subrutinas que manipulan los datos definidos por la clase y, en muchos casos, brindan acceso a esos datos.

Atributo: Los atributos son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Los atributos se guardan en variables denominadas de instancia.

RESUMEN

Este trabajo es la actividad 2 de lenguaje de programación, se esta trabajando el lenguaje de java donde se aprende la programación orientada a objetos.

siguiendo la linea de explicaciones del profesor y enfocados en la creación de clases, métodos, atributos y sus herencias se la tratado de dar solucionó a esta actividad apoyándome de los documentos de unidades de estudio y los encuentros sincrónicos para comprender, analizar y ejecutar una posible solución a los problemas propuestos.

PALABRAS CLAVE: Métodos, clases, instancias, programación orientada a objetos, POO, polimorfa.

# INTRODUCCIÓN

En la actividad numero 2 veremos el desarrollo de los ejercicios propuestos por el profesor, siguiendo con la linea de trabajo de las unidades de estudio y basándose en las mismas se aplico los conocimientos de la programación orientada a objetos en la creación de clases, atributos, métodos, herencias y se trabajaron modificadores de acceso y sus diferentes implementaciónes dentro del código.

# 2. OBJETIVOS

## 2.1 OBJETIVO GENERAL

Conocer, entender e implementar las diferentes clases, atributos y métodos de la programación orientada a objetos.

## 2.2 OBJETIVOS ESPECÍFICOS

* Comprender la herencia de métodos en la POO.
* Conocer y aplicar los modificadores de acceso.
* Establecer instanciamiento de clases.

# 3 PLANTEAMIENTO DEL PROBLEMA

## 3.1 DEFINICIÓN DEL PROBLEMA

**EJERCICIO 1**

Construya una clase llamada Asignatura que represente el nombre de una asignatura y la nota correspondiente obtenida. Se debe implementar:

* Constructor que acepte como parámetros el nombre de la asignatura y la nota
* obtenida.
* Métodos para modificar la nota (setNota()) y para consultar la nota
* (getNota()).
* Método que nos devuelva "Aprobado" si la nota es mayor o igual a 3.5 o
* "Reprobado" si la nota es menor que 3.5.
* Método para consultar el nombre de la asignatura.

Ahora implemente la clase Alumno que incluya referencias a máximo cuatro

Asignaturas a las que el alumno ha asistido. Además de incluir los atributos

nombre y edad. Las operaciones disponibles sobre el alumno son:

* Constructor que acepte como parámetro el nombre del alumno y edad.
* Métodos para modificar el nombre (setNombre) y para consultarlo (getNombre).
* Métodos para modificar y consultar la edad.
* Método que nos devuelva el promedio del alumno.
* Método para agregar una Asignatura a su plan de estudio; verificar que la asignatura no exista previamente las asignaturas de ese Alumno.
* Implemente la clase Aplicación para hacer uso de las clases Alumno y Asignatura.
* Crear 3 alumnos (tres instancias de la clase Alumno) con sus respectivos nombres y edad. Para cada alumno establecer sus asignaturas y la nota obtenida.

Imprimir en pantalla:

* Nombre del alumno.
* Edad.
* Asignaturas que cursó
* Nombre de la asignatura.
* Nota obtenida.
* Si es una asignatura aprobada o no.
* Promedio del alumno

**EJERCICIO 2**

Defina una clase en Java llamada Automóvil, la cual deberá contener los métodos:

arrancar, acelerar y frenar con modificador de acceso público y los datos o

atributos color y modelo con acceso privado.

Utilizando la herencia cree una clase derivada llamada Ferrary que herede de

Automóvil.

Cree instancias de Ferrary que hereden las funciones de Automóvil.

## 3.2 JUSTIFICACIÓN

Estos dos problemas anteriores deben ser solucionados para mejorar los conocimientos en la programación orientada a objetos, las herramientas de estudio como fueron los encuentros sincrónicos y las unidades de estudio fueron de gran utilidad para el desarrollo de estos problemas.

5. MATERIALES Y MÉTODOS

## 5.1 MATERIALES

todo fue desarrollado en IntelliJ ide.

## 5.2 METODOLOGÍA

Programación orientada a objetos, creación y herencia de métodos entre clases.

# 6 DESARROLLO DEL PROYECTO

Aprendí como instanciar objetos, hice repaso de creación de arreglos, fue un reo para mi el primer punto porque me cuesta el tema de instanciamientos y nombramiento correcto de datos.

## 6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO

Creo que en un análisis me fue bien, los códigos al final imprimen lo que se solicita, se usa las clases, sus métodos y atributos, en el punto se usa overwrite, me siento contento con el código logrado.

# CONCLUSIONES

* Aprendí como usar los modificadores de acceso dentro de sus propias clases o fuera de estas según la característica de estos.
* logre usar de forma satisfactoria herencia de métodos en otra clase y métodos propios.
* se logra instanciar clases de forma correcta implementando los conocimientos obtenidos en clase.