

REQUERIMIENTOS MEMORIA ESCRITA PRÁCTICA 1

ASIGNATURA

Programación orientada a objetos

PROFESOR

Jaime Alberto Guzmán Luna

TIPO DE MATERIAL

Material para uso del uso docente

RECUERDE LEER MUY BIEN EL MATERIAL

Universidad Nacional de Colombia

Sede Medellín

2024

Índice

1. Portada	2
2. Descripción general de la solución.	3
3. Descripción del diseño estático del sistema en la especificación UML.	4
4. Descripción de la implementación de características de programación orientada a objetos en el proyecto.	5
5. Descripción de cada una de las 5 funcionalidades implementadas.	6
6. Manual de usuario.	7

1. Portada

Se debe desarrollar una portada que incluya los siguientes elementos:

- **Nombre actividad:** Práctica 1
- **Nombre proyecto:** Agregar el nombre de su proyecto
- **Materia:** Programación Orientada a Objetos
- **Grupo:** Agregar el número de su grupo (01 - 02)
- **Equipo:** Número de su equipo
- **Integrantes:** Agregar el nombre completo de los estudiantes de su equipo
- **Profesor:** Jaime Alberto Guzmán Luna
- **Nombre de la universidad:** Universidad Nacional de Colombia - Sede Medellín
- **Fecha:** 2024

2. Descripción general de la solución.

El análisis del dominio del proyecto es el primer paso en el desarrollo de la aplicación. En esta fase, se identifican y definen los requisitos y necesidades del proyecto. Este análisis incluye:

- **Identificación de los Problemas y Necesidades:** Se deben identificar claramente los problemas que la aplicación pretende resolver y las necesidades específicas de los usuarios y del sistema.
- **Requisitos Funcionales y No Funcionales:** Se deben listar y describir todos los requisitos funcionales (qué debe hacer el sistema) y no funcionales (cómo debe ser el sistema, en términos de rendimiento, seguridad, etc.).

3. Descripción del diseño estático del sistema en la especificación UML.

En esta sección, se debe incluir el diagrama de clases que representa el dominio de la aplicación. Este diagrama debe mostrar todas las clases principales del sistema junto con las relaciones de cardinalidad entre ellas. La cardinalidad indica cuántas instancias de una clase pueden estar asociadas con una instancia de otra clase, proporcionando una visión clara de las interacciones y dependencias dentro del sistema.

Además del diagrama de clases, es importante proporcionar una breve descripción de la función de cada una de las clases en el proyecto. Esta descripción debe incluir:

- **Nombre de la Clase:** El identificador de la clase
- **Propósito:** Una breve explicación del rol y la responsabilidad de la clase dentro del sistema.
- **Relaciones:** Las relaciones de cardinalidad con otras clases, indicando cómo se conecta e interactúa con ellas.

4. Descripción de la implementación de características de programación orientada a objetos en el proyecto.

En el enunciado de la práctica en el ítem 6, se solicitó la implementación de varias características de programación orientada a objetos. Para cumplir con este requerimiento, se debe realizar lo siguiente:

- **Captura de pantalla:** Agregue una captura de pantalla por cada criterio solicitado. Cada captura debe mostrar claramente la implementación del criterio en el código.
- **Ubicación:** Indique la ubicación exacta en el código donde se encuentra la implementación de cada criterio. Esto incluye el nombre del archivo y la línea específica o el método donde se puede encontrar la implementación.
- **Uso:** Describa la utilidad de la implementación de cada criterio en la aplicación. Explique cómo y por qué se utiliza el criterio de programación orientada a objetos en el contexto del proyecto, destacando los beneficios y mejoras que aporta al sistema.

5. Descripción de cada una de las 5 funcionalidades implementadas.

- **Descripción de la funcionalidad:** Proporcione una descripción detallada y completa del comportamiento de cada funcionalidad. Asegúrese de explicar claramente cada una de las interacciones que componen la funcionalidad, describiendo cómo los diferentes componentes del sistema colaboran para lograr el objetivo de la funcionalidad.
- **Diagrama de interacción:** Incluya una imagen del diagrama de interacción correspondiente a cada funcionalidad. Asegúrese de que el diagrama sea claro y permita observar el comportamiento general de la funcionalidad, destacando las interacciones clave y los flujos de datos entre los componentes del sistema.

6. Manual de usuario.

Desarrolle un Manual de Usuario completo que permita guiar a cualquier usuario en el uso de su aplicación. Este manual debe estar diseñado de manera que sea comprensible para personas con diferentes niveles de conocimiento tecnológico.