**Paso a paso del proyecto análisis y diseño orientada a objetos**

**Fase 1:** Análisis y Planificación

1. Lectura y Entendimiento del Enunciado:

* Lee el enunciado para comprender la magnitud del proyecto y su alcance.
* Identifica las especificaciones y requerimientos establecidos por el cliente.

2. Identificación de Clases:

* Analiza el enunciado para identificar las clases necesarias para el proyecto.
* Determina cuántas clases se requieren y cuáles son.

3. Relaciones y Cardinalidad entre Clases:

* Identifica las relaciones entre las clases y su cardinalidad (uno a uno, uno a muchos, muchos a muchos).

\* Considera la multiplicidad de las relaciones (una clase puede tener múltiples relaciones con otras).

* Ejemplo: Si tienes las clases "Empresa" y "Empleado", la relación es que "Empleado trabaja en/para Empresa".

**Fase 2:** Diseño UML

4. Modelado UML:

* Utiliza una herramienta UML para crear un diagrama que represente las clases y sus relaciones.
* El diagrama UML facilita una perspectiva visual clara de la estructura del proyecto.

5. Detalle de Propiedades en UML:

* Agrega las propiedades (atributos) a cada clase en el diagrama UML.
* El diagrama UML muestra las clases, sus relaciones y propiedades.

**Fase 3**: Implementación en NetBeans

6. Creación del Proyecto:

* Abre NetBeans y crea un nuevo proyecto.
* Crea dos paquetes dentro del proyecto.

7. Paquete Principal y Archivo "Modelo":

* En el primer paquete (creado por defecto), crea una clase principal.
* Crea un archivo llamado "Modelo" en este paquete.
* El archivo "Modelo" contendrá el código generado por el UML para las clases.

8. Segundo Paquete para Clases:

* Crea un segundo paquete para contener todas las clases identificadas en el paso 2.

9. Copia de Clases desde "Modelo":

* Copia el código de cada clase desde el archivo "Modelo".
* Pega el código en sus respectivas clases en el segundo paquete.

10. Modificación de Accesos e Importaciones:

* Agrega modificadores de acceso (public, private, protected) a las propiedades de todas las clases.
* Importa las clases necesarias (ej., "Date", "List") para evitar errores de compilación.

**Fase 4:** Finalización de Clases

11. Creación de Constructores:

* Agrega constructores a cada clase.
* Cada clase debe tener un constructor por defecto (sin parámetros).
* Luego, agrega un constructor con los parámetros respectivos a cada clase.

12. Clases Completas:

* Con la creación de los constructores, las clases del proyecto están finalizadas.