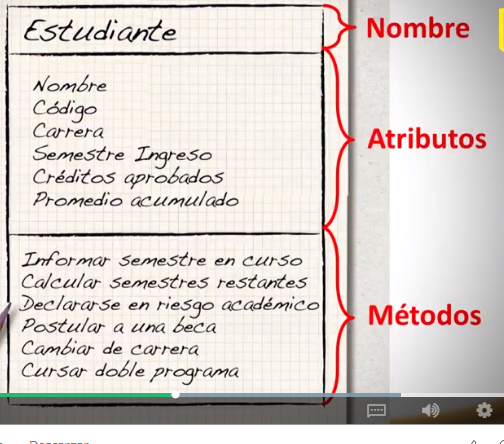
1. Orientación a objetos
   1. Se divide la solución
   2. Se agrupan y asignan las tareas
   3. Colaboran los grupos en la solución
      1. De manera organizada
2. Clases

|  |  |
| --- | --- |
| Nombre | Perro |
| Atributo | Color, tamaño, edad |
| Métodos | Correr, saltar, dormir |



* 1. Formadas por objetos
  2. Definen la conducta y atributos de sus objetos
  3. Pruebas unitarias
     1. Prueba del objeto para ver si funciona
  4. Superclases
     1. Agrupaciones dentro de las clases
  5. Pruebas de integración
     1. Prueban todos los objetos de la clase
  6. Acceso
     1. La clase decide qué o quién tiene acceso a los atributos o servicios de sus objetos
     2. Encapsulamiento
        1. Aislamos objeto
        2. Acceso sólo mediante mensajes
        3. Se puede acceder a ellos desde otra clase
  7. Polimorfismo
     1. Dentro de una clase, dos objetos pueden dar diferente salida al mismo mensaje

1. Objeto
   1. El ejemplar de una clase
   2. Se cea con instrucción new
   3. Atributo
      1. Lo que usa el objeto para realizar la acción
      2. Su contenido
      3. Puede ser información
         1. Números
         2. Caracteres
         3. Símbolos
         4. Variables lógicas
   4. Método
      1. Descripción de lo que un objeto sabe hacer
      2. Define acción (muestra\_Disco)
   5. Herencia
      1. Raíz
         1. La característica que se comparte
   6. Mensajes
      1. Solicitudes de servicio entre objetos
2. JAVA / Estructura
   1. Primero/ Definir las clases
   2. Segundo/Construir objetos
   3. Tercero/ Determinar interacción entre objetos desde un método principal
      1. Método principal/ “main”- Dentro de una de las clases definidas
3. Análisis del problema
   1. Requerimiento funcional
      1. Se describe a través de
         1. Identificador y nombre (R1: Calcular IVA)
         2. Resumen de la operación (Cálculo del IVA del importe indicado)
         3. Entradas de datos para realizar la operación (Indicar base de cálculo)
         4. Resultado esperado (IVA calculado sobre saldo indicado)
      2. Tiene que ser
         1. Una modificación de un valor en el mundo del problema
         2. El cálculo de un valor
         3. Una mezcla de ambos (1.2.)
      3. Aconsejable
         1. Identificador tipo R1
         2. Explicación en infinitivo y fácil de entender
      4. Verificar
         1. Esta operación es una de las razones por las cuales el cliente necesita construir el programa?
         2. La operación no es ambigua? Se entiende?
         3. La operación tiene comienzo y fin?
   2. Mundo / Contexto
   3. Requerimiento no funcionales
4. Diseño de la solución
   1. Detallar mediante lenguaje las características de la solución
      1. Es parte integral de la solución
   2. Construcción de la solución
      1. Implementar programa a partir del diseño y probarlo
      2. Contar con ambiente de desarrollo
         1. Ayuda a encontrar errores de sintaxis
   3. Partes
      1. Diseño
      2. Programa
      3. Pruebas de corrección