

UML Diagrama de clases	Sintaxis de Java
Definición de Clase	
	<pre>public class Empleado { private String nombre; private String apellido; private double sueldo; private int id; private static int num = 0; public Empleado(String pNom, String pApell) { nombre = pNom; apellido = pApell; sueldo = 1000; num++; id = num; } public void escribirNombre() { System.out.println(this.nombre); } // Se puede utilizar desde la clase public static void escribirNum() { System.out.println(num); // No se puede utilizar THIS } ... }</pre>
Definición Tipo enumerado	
	<pre>public enum Vaso{ JARRA(500), TUBO(250), TERCIO(333), CAÑA(200); ... }</pre>
Definición de Interface y Clase Abstracta	
	<pre>public interface Closable{ public void close(); //no existe implementación } public abstract class AbstractClass{ ... }</pre>
Relación de Herencia	
	<pre>public class Item { ... } public class CD extends Item{ ... } public class DVD extends Item{ ... }</pre>
Definición de Interface y Relación de Herencia	
	<pre>public class MyClass implements Closable { ... public void close { ... } } public interface NetworkChannel extends Closable{ ... }</pre>

UML Diagrama de Clases y Objetos	Sintaxis de Java
<p>Diagrama de clases (modelo de base de datos de objetos multimedia)</p>	
<p>Diagrama de objetos para el modelo anterior</p>	<pre> public class Programa { public static void main(String[] args){ Database database = new Database(); CD cd1 = new CD(...); database.addItem(cd1); DVD dvd1 = new DVD(...); database.addItem(dvd1); ... } </pre>
UML Diagrama de secuencia	Sintaxis de Java
<p>Diagrama de clases (modelo de reloj digital)</p>	<pre> public class ClockDisplay{ private NumberDisplay hours; private NumberDisplay minutes; private String displayString; public ClockDisplay() { hours = new NumberDisplay(24); minutes = new NumberDisplay(60); updateDisplay(); } private void updateDisplay() { displayString = hours.getDisplayValue() + ":" + minutes.getDisplayValue(); } ... } public class NumberDisplay{ private int limit; private int value; public NumberDisplay(int rollOverLimit) { limit = rollOverLimit; value = 0; } ... } </pre>
<p>Diagrama de secuencia (usando el modelo anterior) modelando la interacción</p>	<pre> public class Simulador { private ClockDisplay myDisplay; public Simulador(){ myDisplay = new ClockDisplay(); } } public class Programa { public static void main(String[] args){ Simulador simula = new Simulador(); } } </pre>