

IMPORTANTE: se deberán entregar capturas o los archivos XMI, creados con Umbrello, así como los archivos JAVA generados. Se recomienda crear un repositorio en GITHUB y subir el enlace.

1. Instalar Umbrello UML Modeller para construir diagramas UML;

The image shows a Google search for "Umbrello UML Modeller" and a Windows installation window for Umbrello 2.32.0.

Google Search Results:

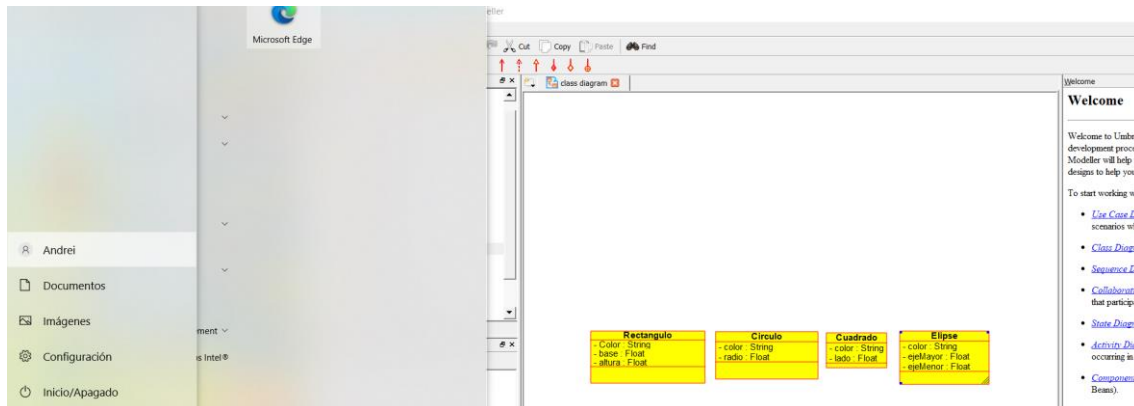
- Search query: Umbrello UML Modeller
- Approximately 31,300 results (0.29 seconds)
- Results include:
 - <https://umbrello.kde.org> - Traducir esta página
 - Umbrello Project - Welcome to Umbrello - The UML Modeller**
 - Umbrello UML Modeller is a Unified Modelling Language (UML) diagram program based on KDE Technology. UML allows you to create diagrams of software and other ...
 - Download**
 - Umbrello is part of KDE Software Compilation which means it ...
 - Manual de Umbrello UML ...**
 - Umbrello UML Modeller ayuda en el proceso de desarrollo de ...
 - Documentation**
 - Umbrello UML Modeller Handbook. The handbook contains ...
 - Features**
 - Most umbrello features and diagram types depends on the ...
 - Más resultados de kde.org »

Windows Installation Window:

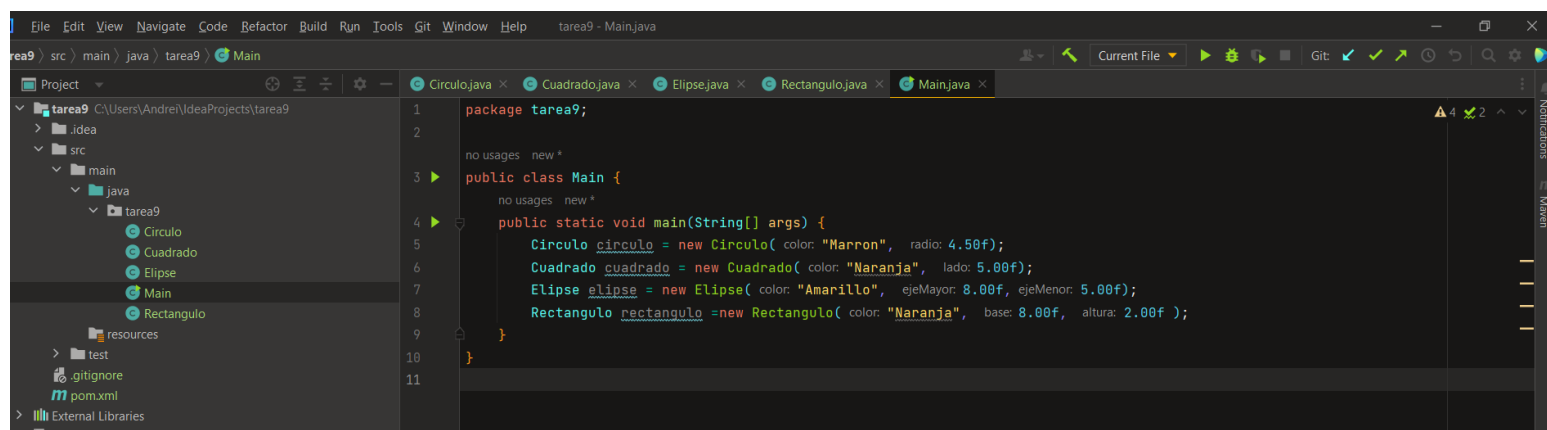
- Title: Instalación de Umbrello 2.32.0
- Icon: Umbrello UML Modeller logo
- Text: **Completando el Asistente de Instalación de Umbrello 2.32.0**
- Text: Umbrello 2.32.0 ha sido instalado en su sistema.
- Text: Presione Terminar para cerrar este asistente.
- Checkbox: ☒ Ejecutar Umbrello 2.32.0
- Buttons: < Atrás, Terminar, Cancelar

2. Para cada figura definir la clase correspondiente

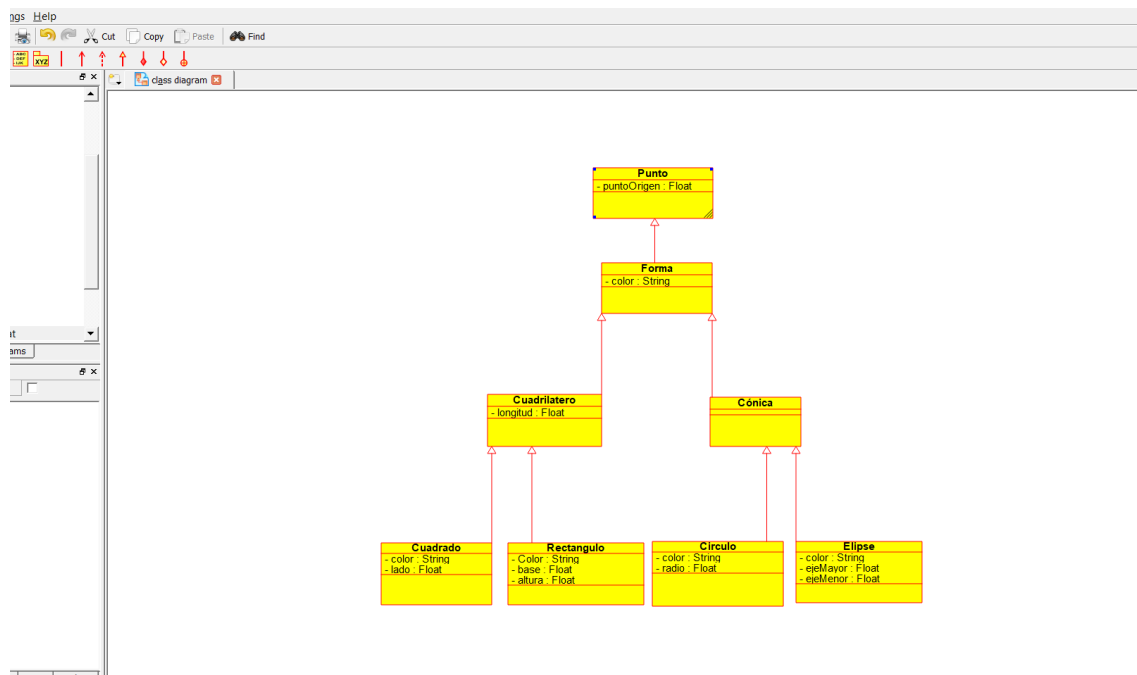
Para cada figura, define el objeto y los valores correspondientes.



3. Generar el código con Umbrello. En Eclipse, crear un proyecto Tarea9 donde vamos a incluir las clases generadas. Crear los constructores y en un método main(), crear un objeto de cada clase



4. A partir de las clases obtenidas en el ejercicio anterior, define una jerarquía de clases usando el concepto de especialización/generalización (incluir la clase Forma, que tendrá el atributo color, las clases cuadrilátero (con el atributo longitud) y cónica (sin atributos), y las clases anteriores que son especialización de las dos últimas e incorporando los atributos que faltan). Crear un objeto Círculo y comprobar que hereda los atributos de Figura y Conica. **Se valorará incluir la definición de una clase Punto para indicar el punto origen de una forma.**



Refactor Build Run Tools Git Window Help tarea9 - Main.java

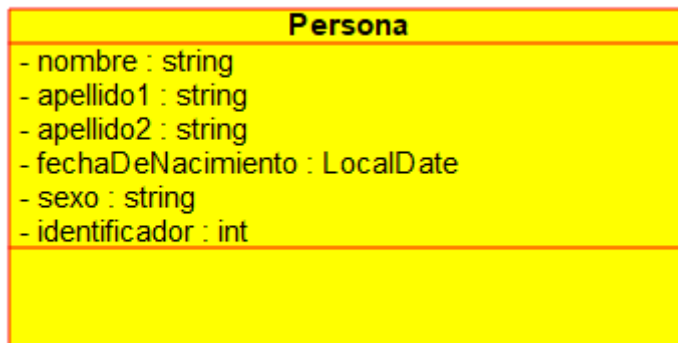
Main

Current File

```
package tarea94;

no usages new *
public class Main {
    no usages new *
    public static void main(String[] args) {
        Circulo circulo = new Circulo();
        circulo.setRadio(1f);
        circulo.setPuntoOrigen(1f);
        Conica conica = new Circulo();
    }
}
```

5. Una persona tiene un nombre, dos apellidos, una fecha de nacimiento, un sexo y un número de identificación. Define la clase y los atributos correspondientes.



6. A partir de la clase **Persona** definida en el ejercicio anterior y del diagrama de objetos, añade asociaciones, cardinalidades y roles para modelar las relaciones de una persona con una clase puesto de trabajo (con atributos nombre, fecha inicio, fecha fin y salario). **Se valorará incluir otros atributos.** Generar el código con Umbrello. Observar cómo se han implementado las asociaciones indicadas en el diagrama. Incluir las clases dentro del proyecto. Crear un ejemplo de una persona que ha tenido dos puestos de trabajo.

