1.- Diseñar una estructura de clases para una aplicación de notas en Kotlin utilizando el patrón Factory Method. Las notas pueden ser de tipo texto, imagen y audio.

2.- La actividad deberá contener mínimamente:

• Diagrama de clases con la estructura del patrón de diseño aplicada a la aplicación de notas.

• Código de las clases sin entrar en detalles de implementación, en la presentación de clases puedes seguir el nivel de detalle del ejemplo del validador de documentos

- **Breve descripción de cada clase**

Diagrama de clases

```
classDiagram

abstract class Nota {
  + titulo: String
  + contenido: String
  + getTitulo(): String
  + getContenido(): String
}

class NotaTexto extends Nota {
  + contenido: String
  + getContenido(): String
}

class NotaImagen extends Nota {
  + imagen: Image
  + getImagen(): Image
}

class NotaAudio extends Nota {
  + audio: Audio
  + getAudio(): Audio
}

class NotaFactory {
  + createNota(tipo: TipoNota): Nota
}

enum TipoNota {
  TEXTO, IMAGEN, AUDIO
}
```

Contenido de las clases

```
abstract class Nota {
  var titulo: String = ""
  var contenido: String = ""

  fun getTitulo(): String {
    return this.titulo
  }

  fun getContenido(): String {
    return this.contenido
  }
}

class NotaTexto : Nota() {
  var contenido: String = ""

  override fun getContenido(): String {
    return this.contenido
  }
}

class NotaImagen : Nota() {
  var imagen: Image = Image()

  override fun getImagen(): Image {
    return this.imagen
  }
}

class NotaAudio : Nota() {
  var audio: Audio = Audio()

  override fun getAudio(): Audio {
    return this.audio
  }
}

class NotaFactory {
  fun createNota(tipo: TipoNota): Nota {
    return when (tipo) {
      TipoNota.TEXTO -> NotaTexto()
      TipoNota.IMAGEN -> NotaImagen()
      TipoNota.AUDIO -> NotaAudio()
    }
  }
}

enum class TipoNota {
  TEXTO, IMAGEN, AUDIO
}
```

4.- Reflexiona sobre cambios futuros:

- ¿Qué pasa si en un futuro se quisiera añadir un nuevo tipo de notas?
- ¿Qué partes de tu aplicación tendrías que modificar?
- ¿Qué nuevas clases tendrías que añadir?

Si en un futuro se quisiera añadir un nuevo tipo de notas, habría que modificar la clase NotaFactory para que pudiera crear instancias de la nueva clase de nota.

Además, habría que añadir una nueva clase que represente el nuevo tipo de nota. Esta clase debería extender la clase Nota y definir el contenido específico de la nota.

En concreto, las modificaciones que habría que realizar son las siguientes:
En la clase NotaFactory:

-Añadir un nuevo caso al método createNota() que devuelva una instancia de la nueva clase de nota.
-Actualizar la documentación de la clase para que refleje la nueva funcionalidad.
En la nueva clase de nota:

-Definir los atributos y métodos que sean necesarios para representar el nuevo tipo de nota.
Implementar las interfaces que sean necesarias para que la nueva clase sea compatible con la clase Nota.