



Programación de Aplicaciones Móviles Nativas

Patrones de Diseño: Factory Method

Javier Ocón Barreiro

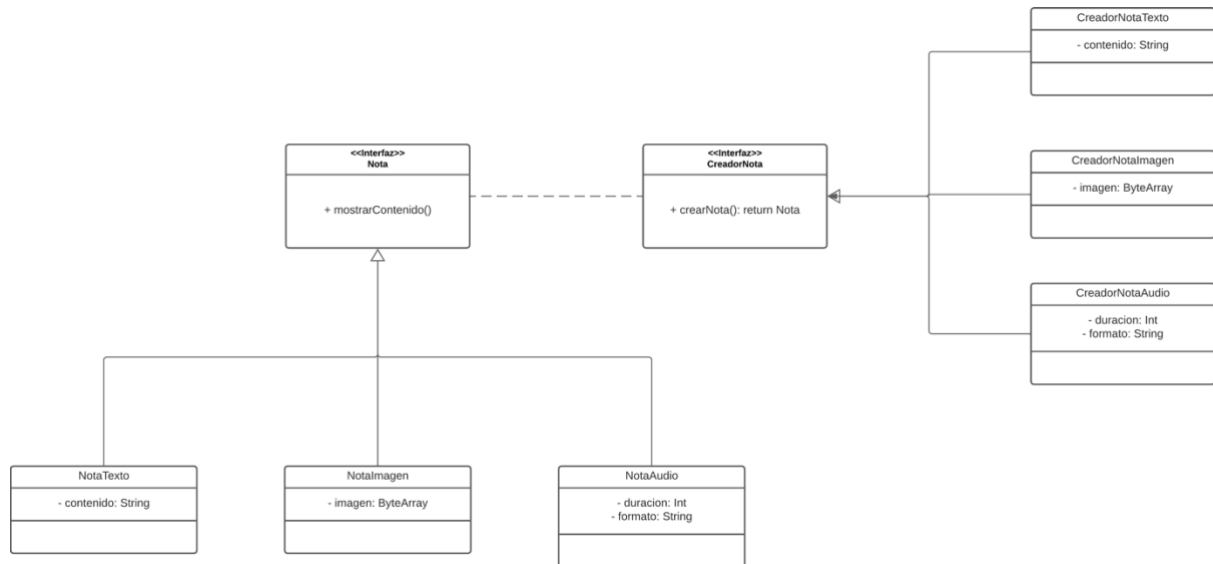


Indice

Table of Contents

<i>Indice</i>	<i>1</i>
<i>Diagrama de Clases</i>	<i>2</i>
<i>Código de las Clases.....</i>	<i>3</i>
<i>Descripción de cada Clase</i>	<i>5</i>
<i>Reflexión sobre Cambios Futuros.....</i>	<i>6</i>
¿Qué pasa si en un futuro se quisiera añadir un nuevo tipo de notas?	6
¿Qué partes de tu aplicación tendrías que modificar?	6
¿Qué nuevas clases tendrías que añadir?	6

Diagrama de Clases



Código de las Clases

Nota.kt

```
1 interface Nota {  
2     fun mostrarContenido()  
3 }
```

CreadorNota.kt

```
1 interface CreadorNota {  
2     fun crearNota(): Nota  
3 }  
4  
5 class CreadorNotaTexto(private val contenido: String) : CreadorNota {  
6     override fun crearNota(): Nota {  
7         return NotaTexto(contenido)  
8     }  
9 }  
10  
11 class CreadorNotaImagen(private val imagen: ByteArray) : CreadorNota {  
12     override fun crearNota(): Nota {  
13         return NotaImagen(imagen)  
14     }  
15 }  
16  
17 class CreadorNotaAudio(private val duracion: Int, private val formato: String) : CreadorNota {  
18     override fun crearNota(): Nota {  
19         return NotaAudio(duracion, formato)  
20     }  
21 }
```

NotaTexto.kt

```
1 class NotaTexto(private val contenido: String) : Nota {  
2     override fun mostrarContenido() {  
3         println("Nota de texto: $contenido")  
4     }  
5 }
```

NotaAudio.kt

```
1 class NotaAudio(private val duracion: Int, private val formato: String) : Nota {  
2     override fun mostrarContenido() {  
3         println("Nota de audio: Duración $duracion segundos, Formato $formato")  
4     }  
5 }
```

◀ Notaimagen.kt

```
1 class NotaImagen(private val imagen: ByteArray) : Nota {  
2     override fun mostrarContenido() {  
3         println("Nota de imagen: [{imagen.size} bytes]")  
4     }  
5 }
```

Descripción de cada Clase

Nota:

La interfaz Nota es la interfaz base que define el contrato común para todos los tipos de notas. Declara el método mostrarContenido(), que debe ser implementado por las clases concretas. Cualquier tipo de nota debe implementar este método para mostrar su contenido.

CreadorNota:

La interfaz CreadorNota es una interfaz de Factory Method que define el contrato para los creadores de notas. Declara el método crearNota(), que las clases concretas deben implementar para crear instancias de notas. Esta interfaz permite crear notas sin conocer los detalles concretos de su implementación.

NotaTexto:

La clase NotaTexto implementa la interfaz Nota. Representa una nota de texto que almacena contenido de texto. La propiedad contenido guarda el texto de la nota, y el método mostrarContenido() muestra el contenido de la nota de texto.

NotaAudio:

La clase NotaAudio implementa la interfaz Nota. Representa una nota de audio que almacena información sobre la duración y el formato del audio. Las propiedades duracion y formato guardan esta información, y el método mostrarContenido() muestra detalles sobre la nota de audio.

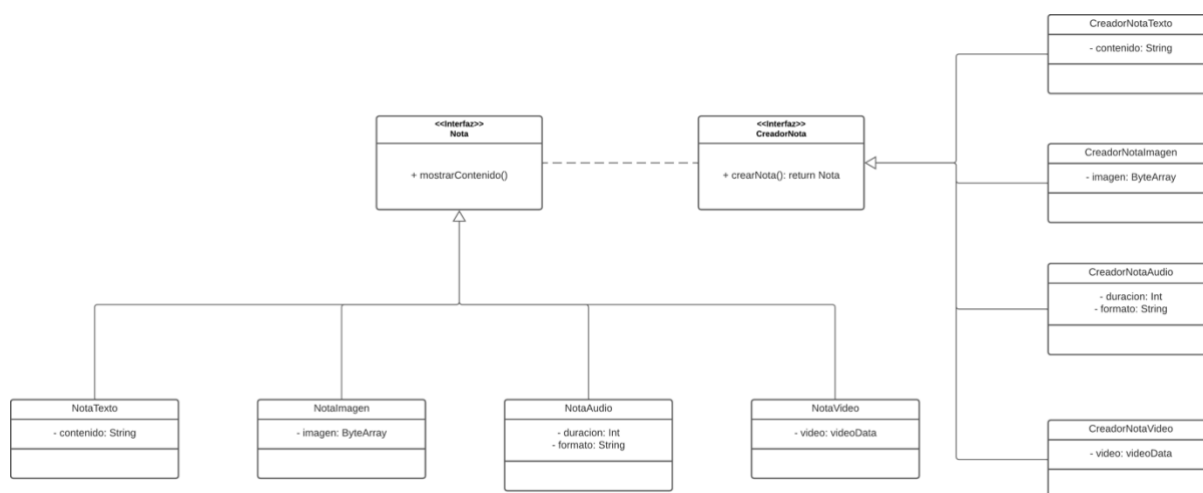
NotaImagen:

La clase NotaImagen implementa la interfaz Nota. Representa una nota de imagen que almacena datos de imagen en forma de bytes. La propiedad imagen almacena los datos de la imagen, y el método mostrarContenido() muestra información sobre la nota de imagen.

Reflexión sobre Cambios Futuros

¿Qué pasa si en un futuro se quisiera añadir un nuevo tipo de notas?

Si en un futuro deseamos añadir un nuevo tipo de notas, por ejemplo **'NotaVideo'**. Esta clase debería al igual que las otras implementar la interfaz **'Nota'** y contener los atributos y lógicas para las notas de vídeo, también deberemos añadir la clase **'CreadorNotaVideo'**. Además de crear las clases, se deberá implementar su creación en la interfaz **'CreadorNota'**, se deberá crear una nueva clase llamada **'CreadorNotaVideo'**, la cual debe implementar de la interfaz **'CreadorNota'** e usar el método **CrearNota()** para crear una nueva instancia **'NotaVideo'**.



¿Qué partes de tu aplicación tendrías que modificar?

Como se dijo en el apartado previo:

Además de crear las clases, se deberá implementar su creación en la interfaz **'CreadorNota'**, se deberá crear una nueva clase llamada **'CreadorNotaVideo'**, la cual debe implementar de la interfaz **'CreadorNota'** e usar el método **CrearNota()** para crear una nueva instancia **'NotaVideo'**.

¿Qué nuevas clases tendrías que añadir?

Se debería de añadir la clase de **NotaVideo** y la clase de **CreadorNotaVideo**.