Vamos a trabajar en la traducción y las diferencias entre Java y Kotlin, algo que va a ser fundamental a medida que avancemos en la programación móvil. Muchos de vosotros estáis acostumbrados a Java, pero como sabéis, en la creación de aplicaciones Android, Kotlin es el lenguaje recomendado.

Hay preparado una serie de ejercicios prácticos que os ayudarán a entender las diferencias clave y cómo traducir el código de Java a Kotlin. Vamos a comparar la sintaxis y funcionalidad de ambos lenguajes, y cómo Kotlin puede simplificar el código.

Ejercicio 1: Declarar variables en Java y Kotlin
En este ejercicio, compararemos cómo se declaran variables en ambos lenguajes y las diferencias en cuanto a la inferencia de tipos.

Java:

```
public class Main {
    public static void main(String[] args) {
        int vida = 100;
        String personaje = "Naruto";
        System.out.println("El personaje " + personaje + " tiene " + vida + " puntos de vida.");
    }
}
```

Explica las diferencias entre el uso de var y val en Kotlin frente a las variables en Java.

Ejercicio 2: Uso de if y operadores ternarios
Vamos a ver cómo traducir el operador ternario en Java a Kotlin, ya que Kotlin no tiene un operador ternario explícito.
Explica cómo funciona la estructura if en Kotlin en comparación con el operador ternario en Java.

Kotlin:

```
fun main() {
    val energia = 80
    val estado = if (energia > 50) "Personaje fuerte" else "Personaje débil"
    println(estado)
}
```

Ejercicio 3: Uso de for y while
Vamos a practicar con bucles, que en ambos lenguajes se usan mucho, sobre todo en situaciones como recorrer inventarios o colecciones de objetos.

Compara el uso del bucle for en ambos lenguajes, haciendo hincapié en cómo Kotlin simplifica la sintaxis.

Java:

```
public class Main {
    public static void main(String[] args) {
        String[] cofres = {"Espada", "Escudo", "Poción"};

        for (String cofre : cofres) {
            System.out.println("Has encontrado: " + cofre);
        }
    }
}
```

Ejercicio 4: Control de flujo con break y continue
Aquí veremos cómo usar break y continue en Kotlin y compararlo con su equivalente en Java.
Explica la diferencia en cómo Kotlin maneja rangos y control de flujo frente a Java.

Kotlin:

```
fun main() {
    for (i in 1..5) {
        if (i == 3) {
            continue // Salta la iteración cuando i es 3
        }
        println("Número: $i")
    }
}
```

Ejercicio 5: Definición de una clase en Java y Kotlin
Vamos a ver cómo crear una clase simple para un personaje con atributos como nombre, vida y ataque.

Explica cómo Kotlin simplifica la definición de clases y el constructor frente a Java.

Java:

```
public class Personaje {
    String nombre;
    int vida;
    int ataque;

    public Personaje(String nombre, int vida, int ataque) {
        this.nombre = nombre;
        this.vida = vida;
        this.ataque = ataque;
    }

    public void mostrarInfo() {
        System.out.println("Nombre: " + nombre + ", Vida: " + vida + ", Ataque: " + ataque);
    }

    public static void main(String[] args) {
        Personaje goku = new Personaje("Goku", 100, 50);
        goku.mostrarInfo();
    }
}
```

Ejercicio 6: Manejo de valores nulos en Kotlin
Kotlin tiene un sistema de tipos nulos más seguro que Java. Explica cómo funciona la seguridad de tipos nulos en Kotlin (String?), comparándola con Java.

Kotlin:

```
fun main() {
    var personaje: String? = null
    if (personaje != null) {
        println("El personaje es $personaje")
    } else {
        println("No hay personaje.")
    }
}
```

Ejercicio 7: Funciones simples y expresiones lambda
Vamos a explorar cómo Kotlin permite simplificar la sintaxis de las funciones y cómo se usan las expresiones lambda.
Compara la simplicidad de las funciones y lambdas en Kotlin con Java, destacando cómo Kotlin es más conciso.

Java:

```
public class Main {
    public static void main(String[] args) {
        System.out.println(multiplicar(5));

        // Lambda en Java (requiere interfaz funcional)
        Operacion operacion = (a, b) -> a + b;
        System.out.println(operacion.sumar(3, 4));
    }

    public static int multiplicar(int num) {
        return num * 2;
    }

    interface Operacion {
        int sumar(int a, int b);
    }
}
```

Ejercicio Final:
Traduce el siguiente fragmento de código Java a Kotlin, que simula un pequeño combate entre dos personajes, donde ambos atacan hasta que uno de los dos quede sin vida.

Java:

```
public class Combate {
    public static void main(String[] args) {
        Personaje goku = new Personaje("Goku", 100, 20);
        Personaje vegeta = new Personaje("Vegeta", 80, 25);

        while (goku.vida > 0 && vegeta.vida > 0) {
            goku.vida -= vegeta.ataque;
            vegeta.vida -= goku.ataque;
            System.out.println("Vida de Goku: " + goku.vida);
            System.out.println("Vida de Vegeta: " + vegeta.vida);
        }

        if (goku.vida <= 0) {
            System.out.println("Vegeta ha ganado el combate.");
        } else {
            System.out.println("Goku ha ganado el combate.");
        }
    }
}

class Personaje {
    String nombre;
    int vida;
    int ataque;

    public Personaje(String nombre, int vida, int ataque) {
        this.nombre = nombre;
        this.vida = vida;
        this.ataque = ataque;
    }
}
```