



CONCLUSIONES DE LOS EJERCICIOS

Marco Antonio Mora Rico

A lo largo de estos ejercicios, hemos recorrido un amplio espectro de características fundamentales y avanzadas de Kotlin, aplicándolas a contextos prácticos y realistas. Desde los conceptos más básicos, como funciones de extensión (`isPositive`, `toLowercaseString`) y propiedades de extensión (`asMiles`), hasta técnicas más sofisticadas como el uso de `let`, `apply` y `also` para encadenar operaciones de forma segura y concisa.

Hemos explorado la programación orientada a objetos con clases abstractas, interfaces y delegación, demostrando cómo crear jerarquías limpias y reutilizables (como `SmartDevice`, `Media`, `PaymentMethod`). La implementación de `data class` y `data object` nos permitió modelar datos inmutables y singletons de manera eficiente, mientras que las clases selladas (`sealed class`) resultaron ideales para representar estados finitos y controlar flujos de manera exhaustiva (`DeliveryStatus`, `Status`).

El manejo de propiedades delegadas (`lazy`, `observable`) nos enseñó a diferir inicializaciones costosas y a reaccionar ante cambios de estado, mejorando el rendimiento y la interactividad de las aplicaciones. También profundizamos en el uso de lambdas con receptor y funciones de orden superior, lo que nos permitió crear DSLs internos y callbacks altamente expresivos (como en `fetchData` y `Button.onEvent`).

Por último, aplicamos operaciones funcionales sobre colecciones (`mapIndexedNotNull`) para transformar datos de forma declarativa y evitar bucles imperativos. En conjunto, estos ejercicios consolidan una base sólida para desarrollar aplicaciones Kotlin robustas, legibles y eficientes, aprovechando al máximo el ecosistema del lenguaje y sus características modernas.