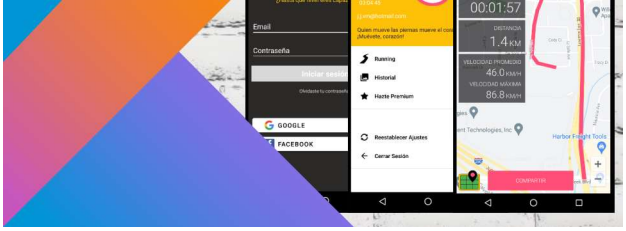

programar aplicaciones con Kotlin

jotajotavm.com/android

TIPOS DE DATOS

Un error curioso que muchos principiantes cometen es creer que una variable podrá almacenar cualquier valor siempre que sea del tipo de dato que se declaró, y NO, no es así.Por ejemplo, si declaras una variable de tipo int en Kotlin puedes almacenar cifras como 24, 8492, o 18284, incluso numero negativos como -204923

Pero **eso no significa que puedas almacenar cualquier numero entero, el espacio reservado es limitado**, en este caso para variables de tipo int en Kotlin se usan 32 bits, por lo que el mayor numero posible a guardar es 2147483647. Si probaras a guardar 2147483648 (es decir, 1 más que el limite) ya no se podría. ¿Por qué? ¿Acaso no sería tal cifra también un número entero? Sí. **Pero tal número requiere de más memoria de la que se ha reservado**, recuerda que todo internamente funciona en binario, ceros y unos, y el modo de almacenar dichas cifras es diferente. Por lo que recuerda que cada tipo de dato tiene unos límites

También debes saber que con la misma cantidad de bits se pueden almacenar unos rangos diferentes de valores en función de si utilizas ese rango para guardar números positivos o, por otro lado, positivos y negativos al mismo tiempo.

Por ejemplo, si para almacenar un número entero disponemos de 4 bytes (32 bits) de memoria tememos que: **4 Bytes = 4x8 = 32 bits** Con 32 bits se pueden representar **232=4294967296 valores**:

- Sólo positivos (enteros sin signo): del 0 al 4294967295
- Positivos y negativos (enteros con signo): del -2147483648 al 2147483647

¿En qué casos tiene sentido definir cifras de este tipo? Bueno hay valores que JAMÁS podrán ser negativos, por lo que se podría **delimitar el tipo de dato para evitar errores**. Por ejemplo la edad de una persona nunca podrá ser un numero negativo. O hay valores que JAMÁS podrán ser negativos, como el número de personas que haya conectadas en un chat.

Así mismo notarás que hay diferentes tipos de números decimales. ¿Por qué? Porque algunos tienen **capacidad para almacenar más parte decimal aumentando la precisión de la cifra**, algo muy útil para cálculos científicos. **De modo que se puede utilizar la misma cantidad de bits de diferentes modos: solo para positivos, para positivos y negativos, para decimales con menor precisión o con mayor precisión.** Pero todo es cuestión de espacio. Así que no creas que puedes guardar cualquier valor en una variable tan solo porque el valor es del tipo de dato de la variable que creaste. También tiene que estar dentro del rango de espacio que dicha variable pueda almacenar

Estos son los tipos de datos numéricos que hay en Kotlin, el numero de bits que usa. Con eso podremos saber el numero de valores que pueda almacenar

Double	64
Float	32
Long	64
Int	32
Short	16
Byte	8