

## Para saber mais - Sobre inicialização com lazy

Aplicamos a estratégia preguiçosa, ou melhor, o *lazy* de delegação de properties para conseguirmos inicializar a `viewDaActivity`, evitando a situação na qual a gente tentava inicializá-la com a property `window.decorView` sendo `null`.

### Outras situações além da que vimos

Porém, além desse tipo de cenário, o *lazy* pode ser útil para **instância que são custosas**, ou seja, que tendem a "demorar para inicializar" e que só serão utilizadas em momentos específicos (que não são no momento que a classe é construída).

Um exemplo comum seria criar uma property para representar um Web Client responsável em realizar requisições HTTP que exige diversas configurações de uma vez, sendo que só será utilizada num clique do usuário.

Em outras palavras, perceba que ela não precisa ser inicializada logo de cara, isso seria processamento jogado fora de maneira...

Considerando cenários como este, o *lazy* pode ser bem útil. Se a sua property não precisa ser utilizada logo de cara, considerar a inicialização por meio do *lazy* tende a ser vantajoso.

### Outras estratégias

Como eu havia comentado, além do *lazy* existem outras técnicas de delegação de properties. Você pode consultá-las diretamente na [documentação do Kotlin \(https://kotlinlang.org/docs/reference/delegated-properties.html\)](https://kotlinlang.org/docs/reference/delegated-properties.html).