Aula 34

Paginação

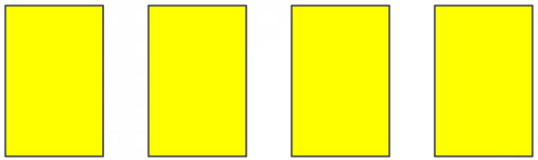
DigitalHouse > Coding School

Paginação - O que é?

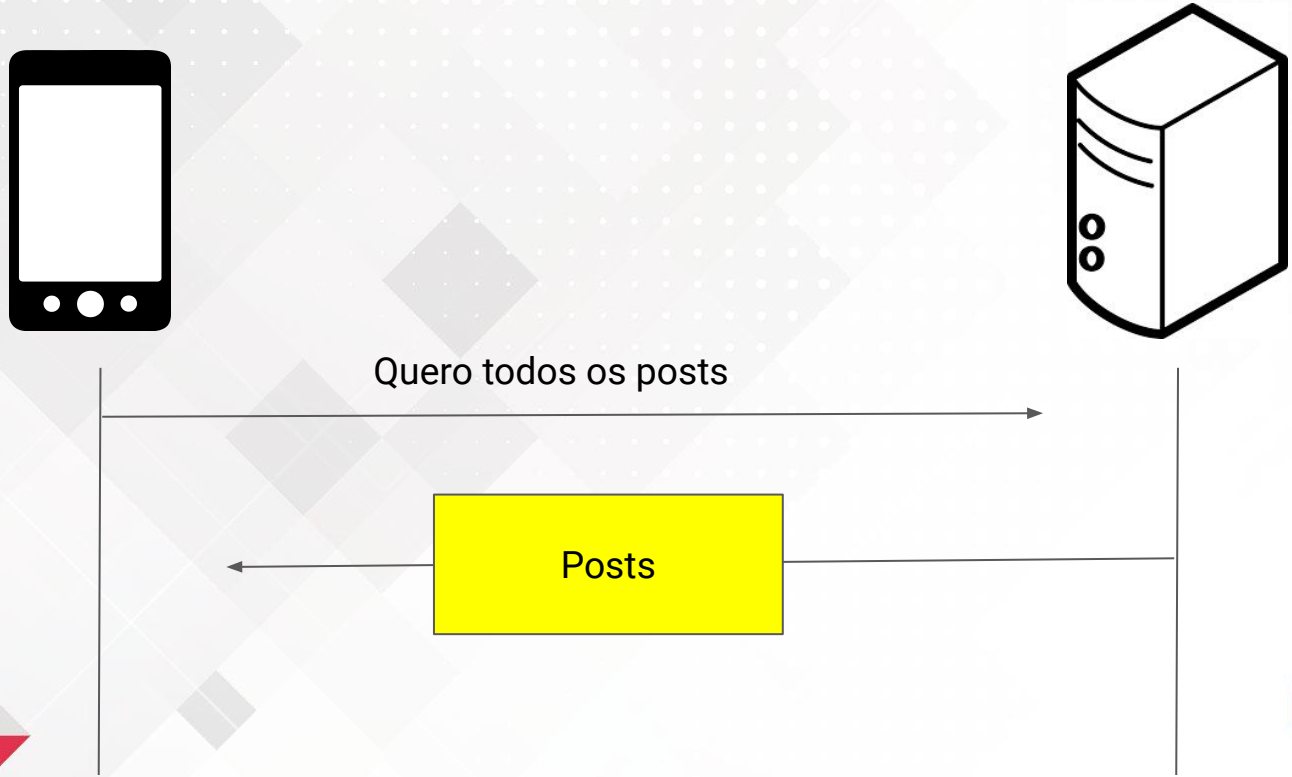
Dividir em partes as informações que pedimos a um serviço, seja por motivos de desempenho, largura de banda ou outros.

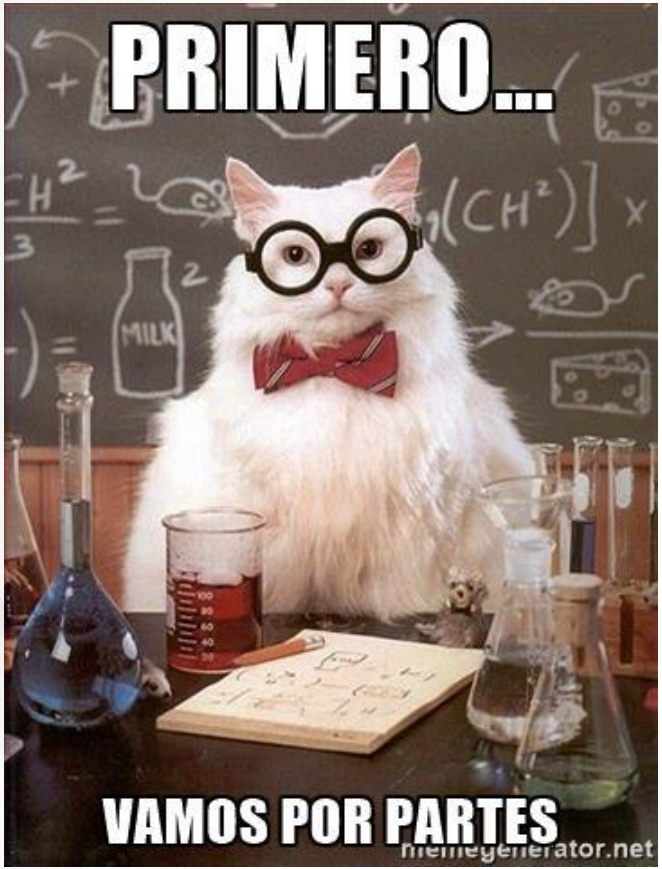
Normal

Paginado



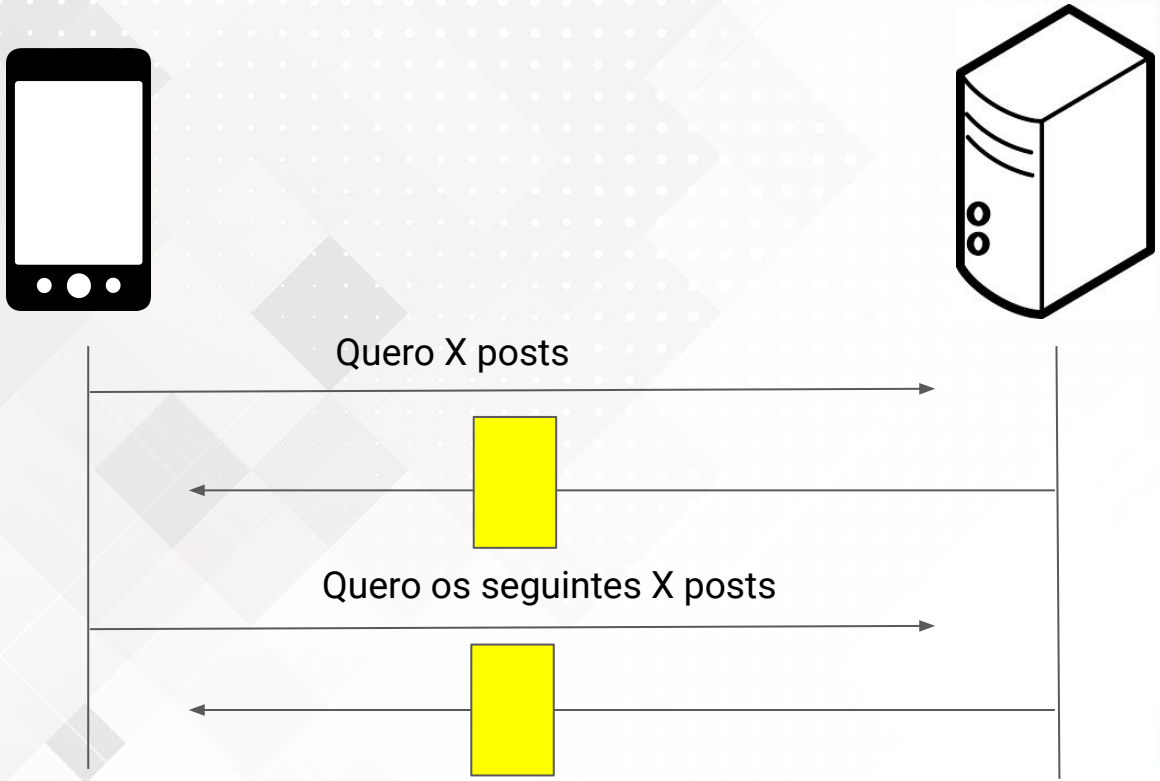
Paginação





DigitalHouse >
Coding School

Paginação



Paginação

- **total -** Quantidade total de registros
- **page -** Número da página atual
- **limit -** Quantidade de registros por página
- **offset -** A partir de que registro começa a página

Exemplo com RickandMortyApi:

Info and Pagination

The API will automatically paginate the responses. You will receive up to 20 documents per page.

Each resource contains an `info` object with information about the response.

Key	Type	Description
count	int	The length of the response
pages	int	The amount of pages ← A quantidade de páginas.
next	string (url)	Link to the next page (if it exists)
prev	string (url)	Link to the previous page (if it exists)

Exemplo com RickandMortyApi:

Agora que sabemos o parâmetro de paginação da API, podemos incluir na chamada do nosso End-Point.

pages	int	The amount of pages

Exemplo com RickandMortyApi:

Para implementar a paginação precisamos adicionar na Interface o parâmetro que encontramos na documentação da API.

```
@GET( value: "character/")
suspend fun getResponseCharacter(
    @Query( value: "page") page: Int = 1): CharacterResponse
```

Exemplo com RickandMortyApi:

Dentro da classe repository adicionamos o parâmetro chamado “page”, como dizia na documentação da API.

```
private var url = "https://rickandmortyapi.com/api/"
private var service = EndPointApi::class

private val serviceRick = RetrofitInit(url).create(service)

suspend fun getCharacterService(page: Int): CharacterResponse = serviceRick.getResponseCharacter(page)
```

Parâmetro adicionado

DigitalHouse > Coding School

Exemplo com RickandMortyApi:

Agora onde de fato a implementação de paginação será realizada. Na nossa “View” criaremos um método chamado setScrollView, ele irá carregar uma nova página de itens no nosso recycler view. Veja o exemplo no próximo slide.

```
private fun setScrollView() {
    recyclerView.run { this: RecyclerView
        addOnScrollListener(object : RecyclerView.OnScrollListener() {
            override fun onScrolled(recyclerView: RecyclerView, dx: Int, dy: Int) {
                super.onScrolled(recyclerView, dx, dy)
                val target : LinearLayoutManager? = recyclerView.LayoutManager as LinearLayoutManager?

                val totalItemCount : Int = target!!.itemCount

                val lastVisible : Int = target.findLastVisibleItemPosition()

                val lastItem : Boolean = lastVisible + 5 >= totalItemCount

                if (totalItemCount > 0 && lastItem) {
                    page++
                    viewModel.getAllCharacters(page)
                }
            }
        })
    }
}
```

Exemplo e exercícios....