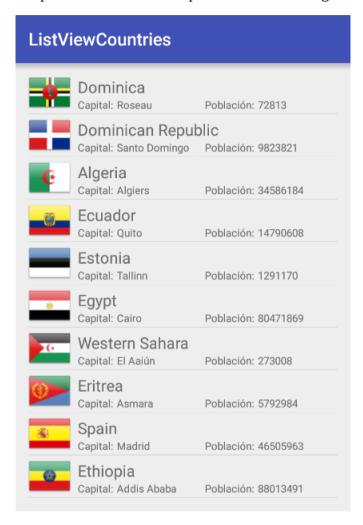
EXERCICI

LISTVIEW AMB VIEWHOLDER

A l'anterior pràctica hem implementat una llista de països similar a la següent:



En el seu moment ja comentarem que podien dur-se a terme una sèrie d'optimitzacions en el mètode getView() seguint el patró de disseny **ViewHolder**.

L'objectiu d'aquesta pràctica és aplicar aquestes optimitzacions al nostre projecte de manera que es faça un ús més responsable de la CPU, la memòria i per tant de la bateria del dispositiu.

Passos a seguir:

- Obrir el **projecte** ListViewCountries realitzat en la pràctica anterior..
- Localitzar el mètode getView() i comprovar si el paràmetre convertView és null. En cas afirmatiu haurem de crear un LayoutInflater per a "unflar" el nostre layout personalitzat, en cas contrari podem "reutilitzar" el layout i ens estalviem tornar-ho a "unflar".
- La següent optimització consistirà en definir la nostra classe estàtica **ViewHolder** per a guardar les referències a l'ImageView i els **TextViews**. D'aquesta forma, ens estalviarem

- cercar els components (findViewById) cada vegada que es mostra un element de la llista, tan sols haurem de fer-lo pel primer element.
- En el mètode getView definirem un objecte de la nostra classe ViewHolder. En el cas que la convertView (item) rebuda siga null, haurem de crear-la i assignar-la mitjançant el mètode setTag(), en cas contrari tan sols haurem d'obtenir la referència al ViewHolder mitjançant el mètode getTag().
- L'últim pas serà assignar els valors de l'ImageView i dels TextView mitjançant el nostre ViewHolder.

Documents a lliurar:

- **MainActivity.java** (Activity principal de l'aplicació)
- Country.java (Model per als països)
- **CountryAdapter.java** (Adaptador personalitzat)
- **CountryParser.java** (Carregador d'objectes Country a partir del xml).
- **activity main.xml** (Layout de l'Activity principal)
- **listitem_country.xml** (Layout per a cada element de la llista)

Es valorarà:

• CountryAdapter (100%)