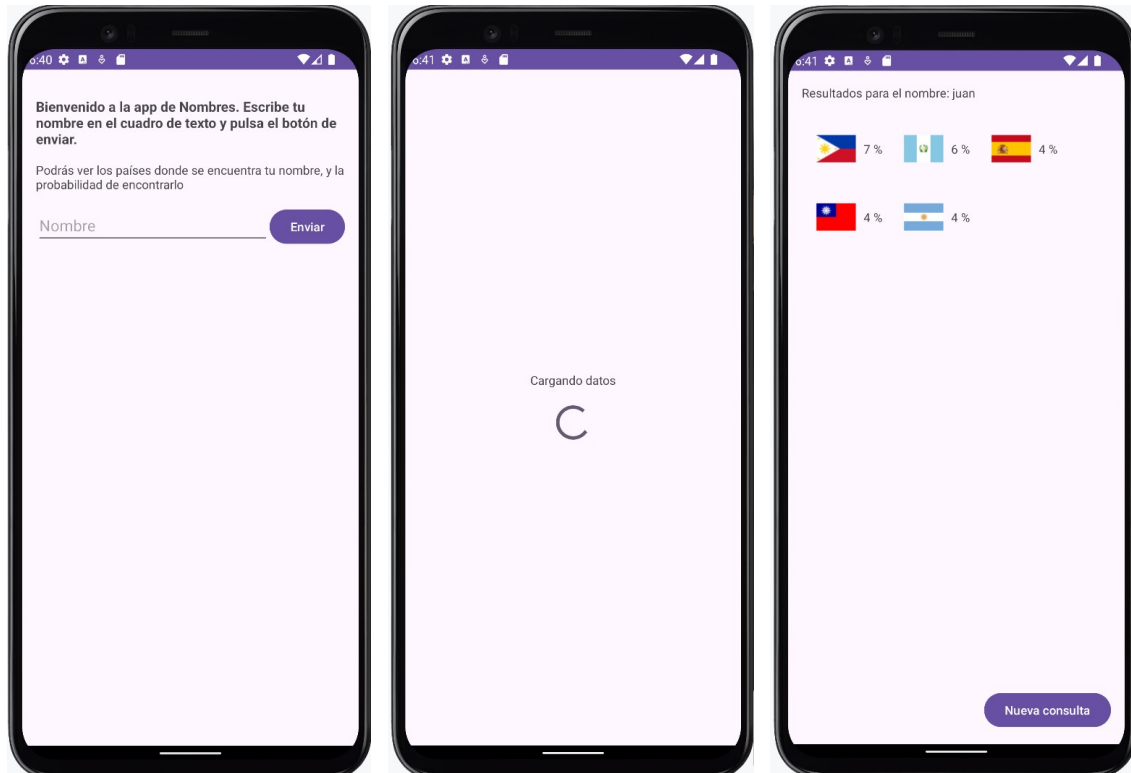


EJERCICIO 8.1

NOMBRES



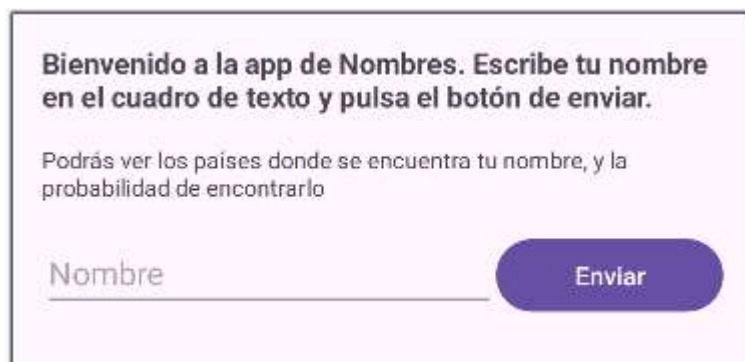
1. - Puesta en marcha del proyecto

- Crea un proyecto en Android Studio
- Concede el permiso de internet a la app
- Habilita view binding, Moshi, Retrofit, Coil y el Navigation component
- Crea los paquetes modelo, vista y viewmodel
- Mueve MainActivity al paquete vista

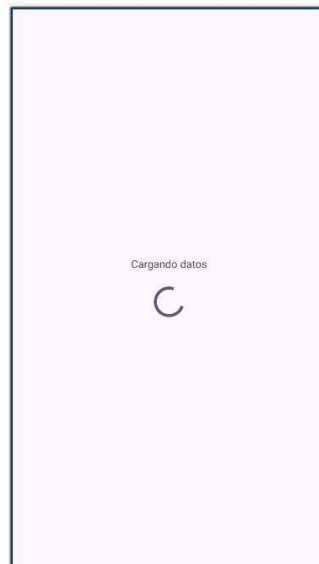
2. - Diseño de la interfaz

- Crea los siguientes tres Fragment, diseñando su interfaz con la vista xml o con el diseñador de Android Studio, como prefieras:

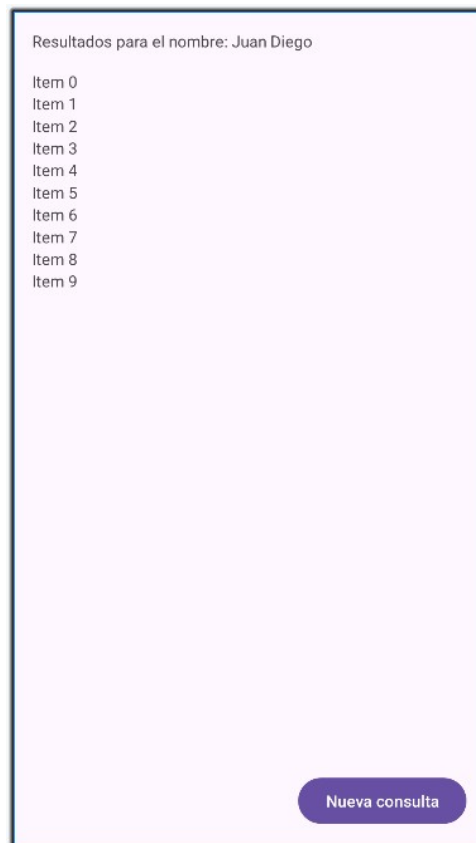
- InicioFragment



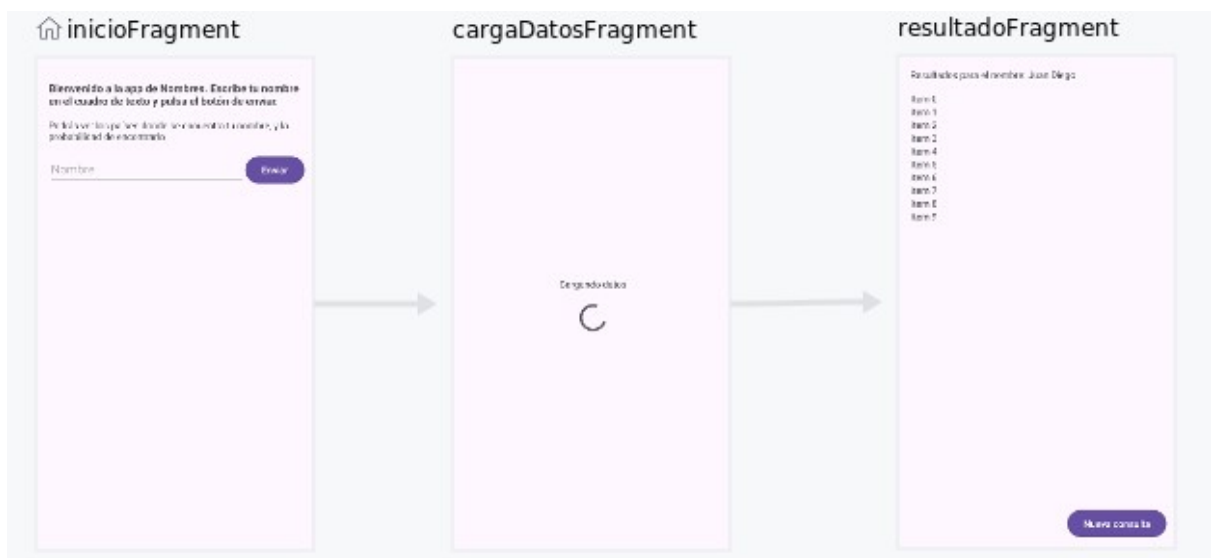
- CargaDatosFragment



- ResultadoFragment



- Crea el siguiente grafo de navegación



- Haz que cargaDatosFragment reciba un argumento de tipo String llamado **nombre**
- Haz que al pulsar el botón de atrás en resultadoFragment, se pase directamente hacia inicioFragment, saltándose cargaDatosFragment
- Haz que MainActivity muestre un FragmentContainerView que contenga el NavHostFragment y navegue según el grafo anterior

3. - Programación de InicioFragment

- Abre el código fuente de InicioFragment
- Inicializa el view binding
- Haz que al pulsar el botón “Enviar” se navegue hacia cargaDatosFragment, pasando como parámetro el texto escrito en el cuadro de texto

4. - Consulta al API de nombres

- Abre tu navegador de Internet y consulta el siguiente api: (sustituye el parámetro nombre escrito en rojo por tu nombre)

`https://api.nationalize.io/?name=nombre`

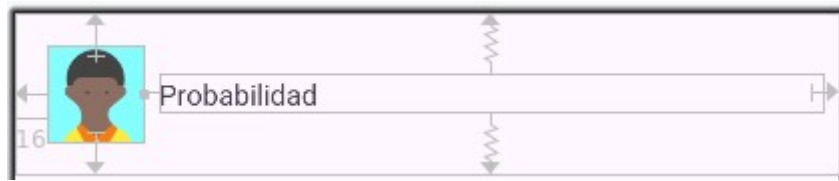
- Observa que el api devuelve un json con los países donde está ese nombre y las probabilidades de ser encontrado
- En el paquete modelo crea las clases que consideres necesarias para que Moshi pueda convertir dicho json en objetos, teniendo en cuenta que en nuestra app necesitaremos el nombre, el identificador del país y la probabilidad del nombre
- Haz que todas las clases que crees en el punto anterior implementen la interfaz Serializable
- Crea una interfaz llamada NombresApi y pon en ella un método para que Retrofit, integrado con Moshi, consulte el api y obtenga los objetos que has creado anteriormente
- Crea una clase llamada NombresRepository, que contenga una variable de instancia de tipo NombresApi. Dicha variable de instancia será inicializada en un bloque init por medio de un objeto Retrofit.
- Añade a NombresRepository un método que reciba como parámetro un nombre y nos devuelva la lista de países y probabilidades donde encontrar ese nombre, usando para ello el objeto NombresApi
- Haz que en el grafo de navegación, resultadoFragment reciba como argumento un objeto de tipo Serializable, cuyo tipo sea la clase que hayas programado para implementar la respuesta del api.

5. - Programación de CargaDatosFragment

- Abre CargaDatosFragment
- Crea un método (vacío de momento) llamado cargarDatos y llámalo dentro de onCreateView
- Programa cargarDatos de forma que lance una corrutina y en su interior obtenga la lista de países y probabilidades correspondientes al nombre que CargaDatosFragment recibe como parámetro de la pantalla anterior. En caso de que todo vaya bien, se navegará hacia ResultadoFragment pasando como parámetro el objeto que da la respuesta del api. Si se produce un error, se mostrará un Toast informando del error y no se hará nada más.
- Abre ResultadoFragment y haz que en onCreateView se muestre en Logcat el objeto de respuesta del api recibido como parámetro. (Cuando veas que funciona lo puedes borrar)

6. - Programación del RecyclerView

- Diseña la siguiente vista para los resultados que mostrará el RecyclerView



- Crea una clase ProbabilidadHolder, que herede de RecyclerView.ViewHolder y que porte un país con su probabilidad, y tenga un método para establecer dicha pareja, actualizando la interfaz. La imagen que se mostrará será la bandera del país, que se obtiene con la siguiente URL:

<https://flagsapi.com/PAIS/flat/48.png>

- Crea una clase ProbabilidadAdapter, que herede de RecyclerView.Adapter<ProbabilidadHolder>, reciba en su constructor la lista de probabilidades (país con su probabilidad) y sobrescriba sus métodos de esta forma:
 - getItemCount: Devuelve el tamaño de la lista de probabilidades
 - onCreateViewHolder: Crea un inflater, lo usa para obtener el binding de un ProbabilidadHolder, y finalmente, devuelve un ProbabilidadHolder con dicho binding
 - onBindViewHolder: Pone en el ProbabilidadHolder recibido como parámetro el país y probabilidad que ocupan en la lista de probabilidades la posición pasada como parámetro

7. - Programación de ResultadoFragmentViewModel

- Abre ResultadoFragmentViewModel
- Añade una variable de instancia que guardará la lista de países y probabilidades que se consulte del api

8. - Programación de ResultadoFragment

- Abre ResultadoFragment
- Añade al método inicializarViewModel una línea para que la lista de países y probabilidades que hay en el view model se rellene a partir de parámetro que recibe ResultadoFragment
- Añade a ResultadoFragment un método llamado inicializarNombre, que ponga al texto de la parte superior el nombre que ResultadoFragment recibe como parámetro
- Añade a ResultadoFragment un método llamado inicializarRecyclerView que le ponga un ProbabilidadAdapter que tenga la lista que hay en el view model
- Haz que al pulsar el botón de nueva consulta se vuelva a la pantalla de inicio