

Documentación Técnica

Decisiones Técnicas

Arquitectura Basada en BLoC/Cubit:

Se eligió utilizar flutter_bloc para la gestión del estado debido a su popularidad y robustez en la comunidad Flutter. La separación de lógica de negocio y UI facilita el mantenimiento y la escalabilidad del proyecto.

La capa de presentación utiliza Cubits para manejar estados simples, lo que simplifica la lógica de gestión del estado y mejora la legibilidad.

Uso de Dio para Comunicaciones HTTP:

Se seleccionó dio como cliente HTTP por su flexibilidad y características avanzadas, como interceptores y cancelación de solicitudes. Esto facilita la gestión de las comunicaciones con la API de Rick and Morty.

Clean Architecture:

La arquitectura limpia fue adoptada para mantener una clara separación de responsabilidades, facilitando la prueba y el mantenimiento del código. Esto permite que cada capa tenga un propósito bien definido.

Desafíos Encontrados

Gestión del Estado Compleja:

Inicialmente, hubo desafíos para garantizar que los estados del Cubit se manejarán correctamente, especialmente en la búsqueda de personajes donde la UI necesitaba actualizarse dinámicamente en función de los resultados de la búsqueda.

Manejo de Errores:

Asegurar que los errores se manejen de manera elegante en toda la aplicación fue un desafío, especialmente para mostrar mensajes de error en la UI cuando las solicitudes a la API fallan.

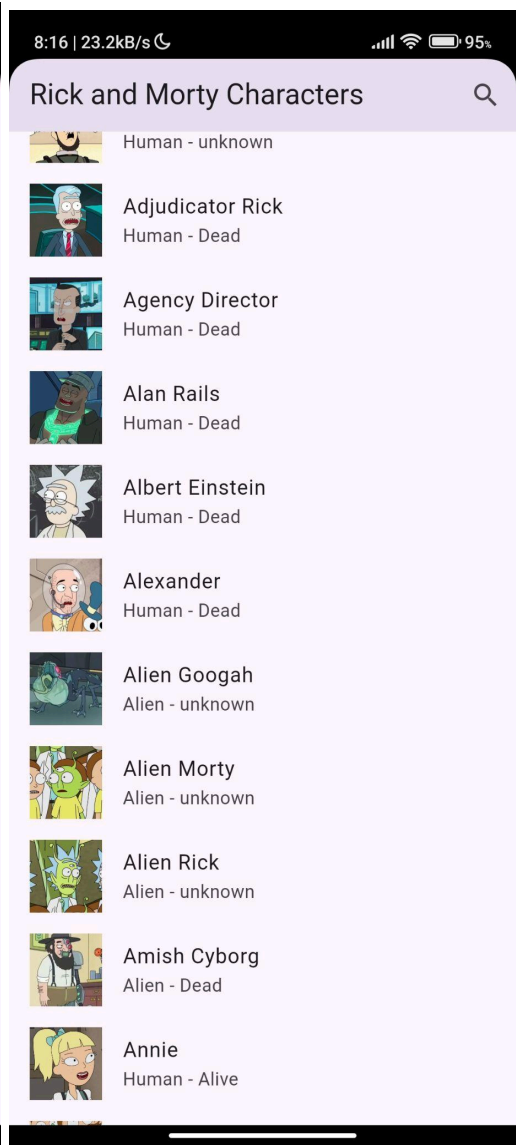
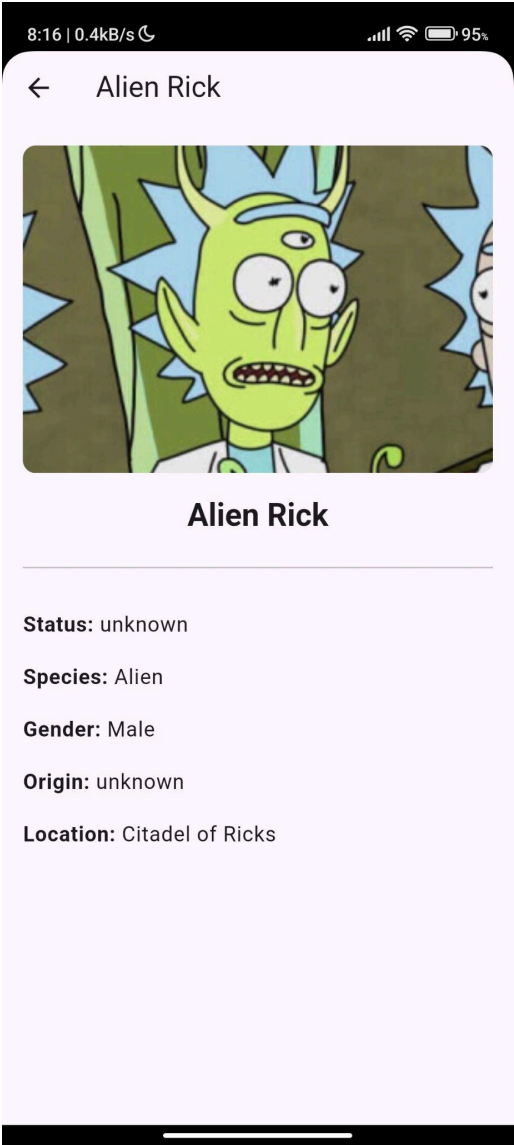
Pruebas Automatizadas:

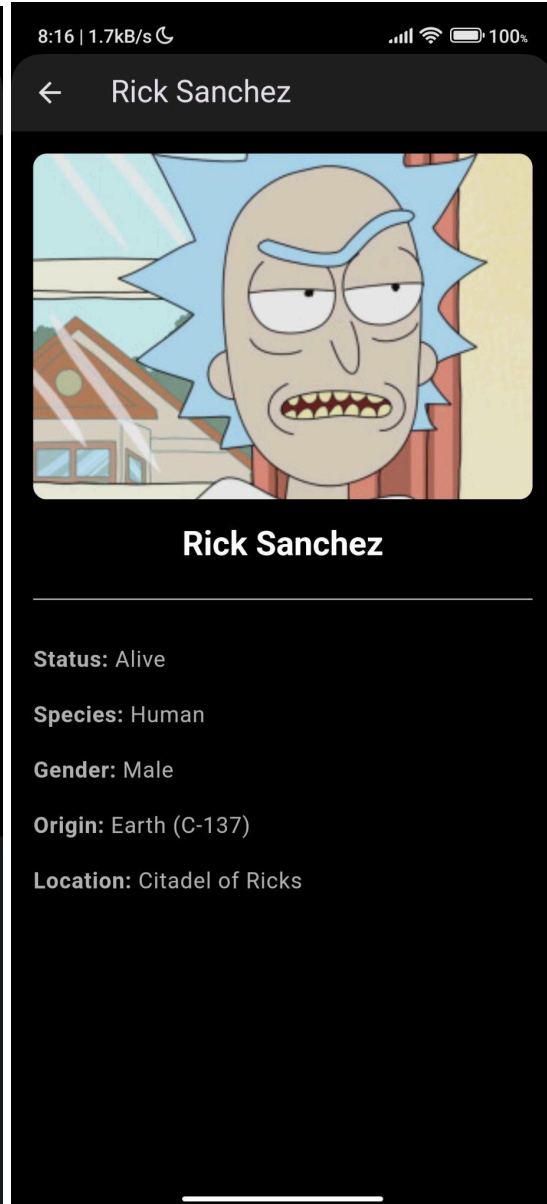
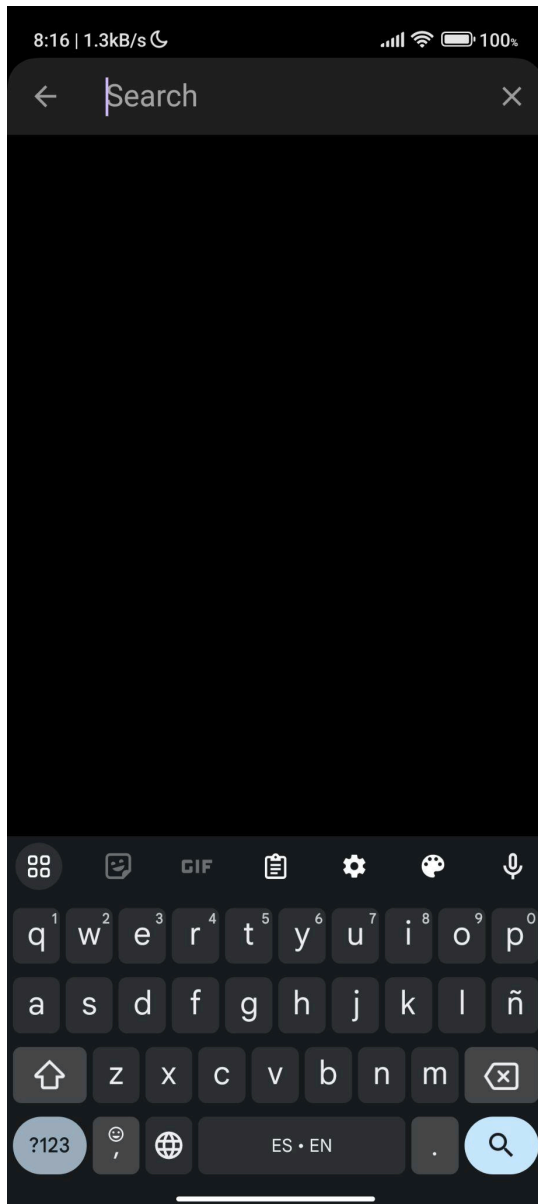
La configuración de pruebas para Cubits, especialmente al manejar estados y asegurar que las comparaciones de estados fueran precisas, requirió un uso cuidadoso de Equatable y ajustes en los mocks utilizando mocktail.

Mejoras:

Durante el desarrollo del proyecto, se identificó la necesidad de mejorar la implementación de las pruebas para asegurar que los estados de los Cubits se manejen y se comparen correctamente. Anteriormente, se encontraron problemas con la comparación de instancias de estado en las pruebas, lo que causaba errores y falsos negativos en los resultados.

PersonajesCubit y sus estados están definidos en un solo archivo. Para mejorar la organización y mantener la claridad del código, se debe separar el *PersonajesCubit* y los estados en archivos independientes.





8:16 | 127kB/s

100%

Rick and Morty Characters



Rick Sanchez
Human - Alive



Morty Smith
Human - Alive



Summer Smith
Human - Alive



Beth Smith
Human - Alive



Jerry Smith
Human - Alive



Abadango Cluster Princess
Alien - Alive



Abradolf Lincler
Human - unknown



Adjudicator Rick
Human - Dead



Agency Director
Human - Dead



Alan Rails
Human - Dead



Albert Einstein