FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

PROGRAMACIÓN IV

INFORME APP CON SQLite

Profesor

Carlos Andrés Guaita Ayala

Autor

Esteban Enríquez

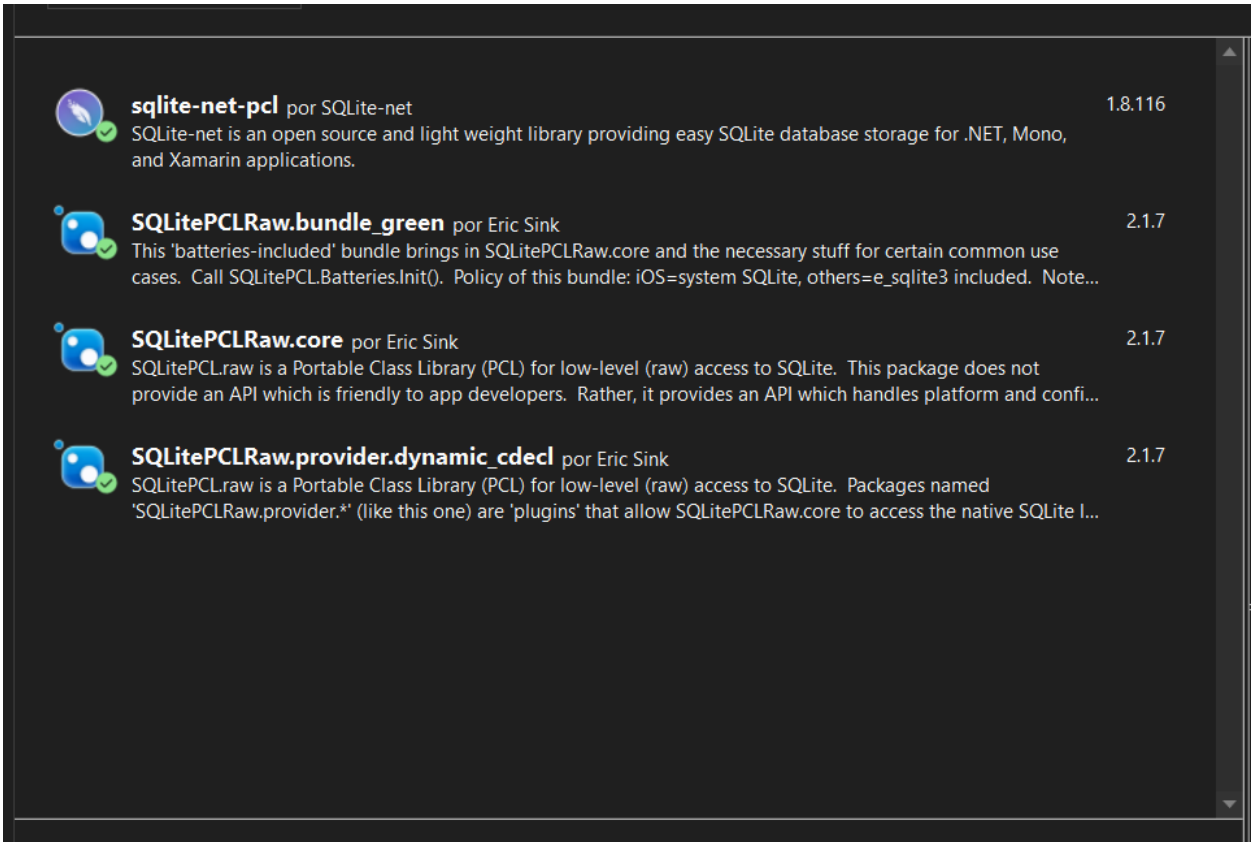
Año

2023

Contexto:

Se creó una aplicación para teléfonos móviles con el uso de .NET MAUI, la cual contiene un CRUD acerca de lo productos, se ha usado SQLite para poder almacenar todos los datos y el modelo MVVM.

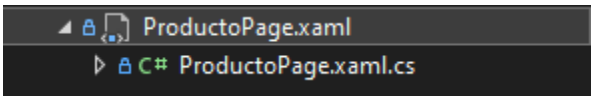
- **Paquetes Nuggets Necesarios para SQLITE**

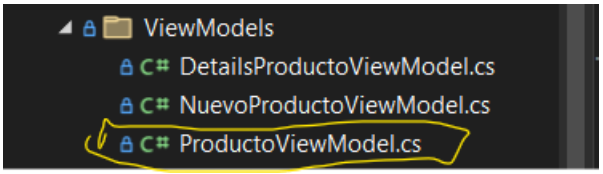


1.Funcionalidad

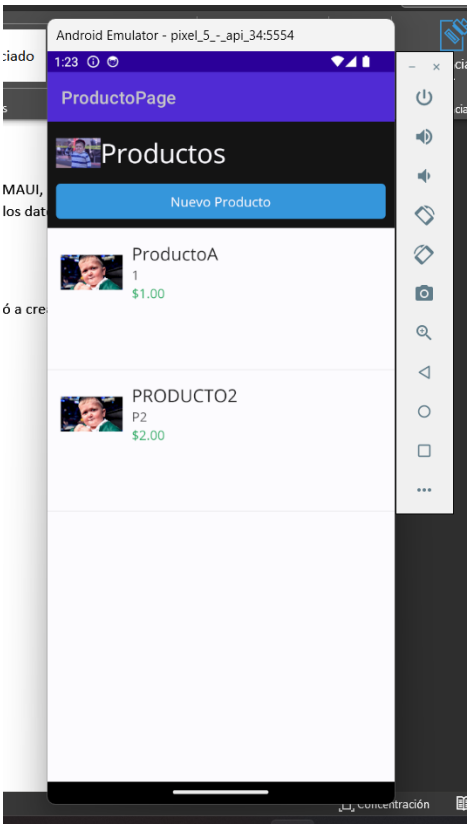
Cargar Productos

En esta página se centrará toda la aplicación, para ello se procedió a crear dentro de este proyecto un elemento XAML el cual se llamará ProductoPage e igual tendrá su viewModel.

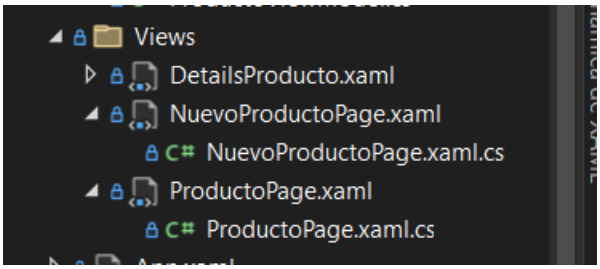


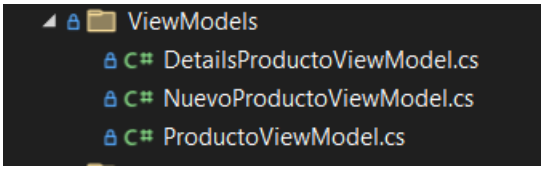


La pantalla que se va a crear es la siguiente:

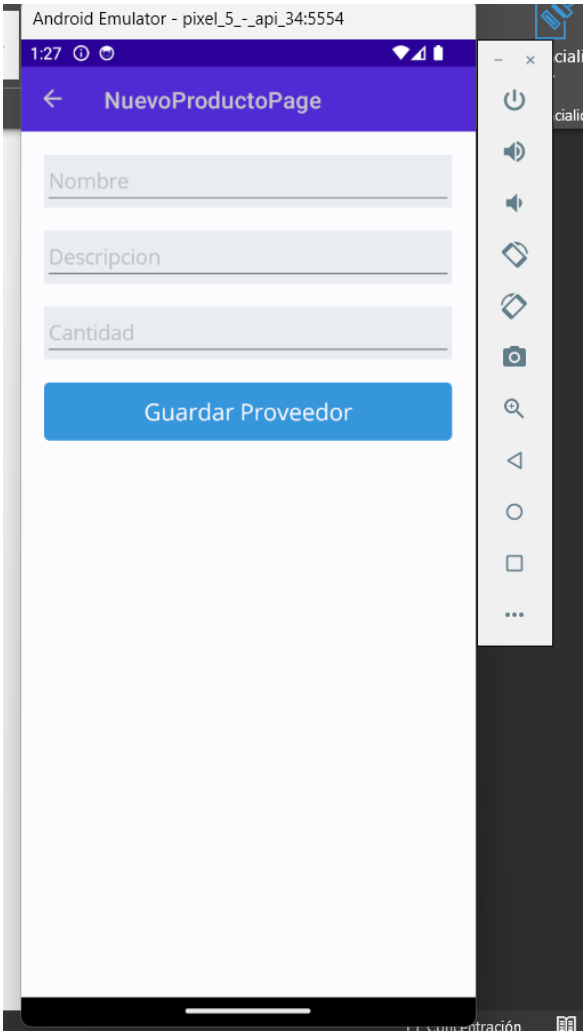


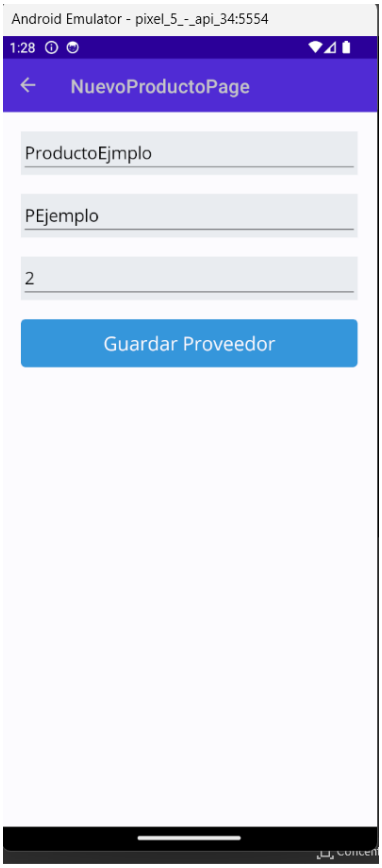
Nuevo Producto

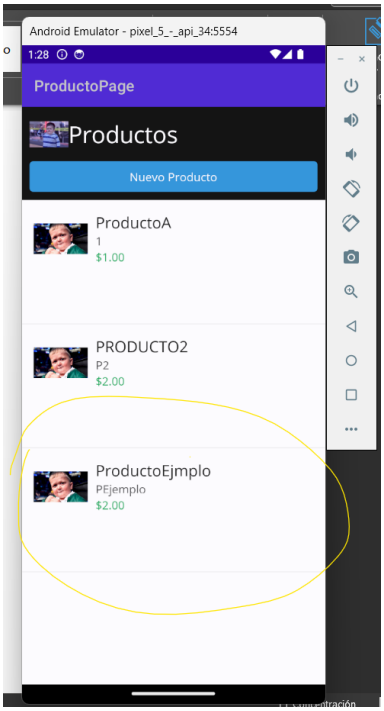




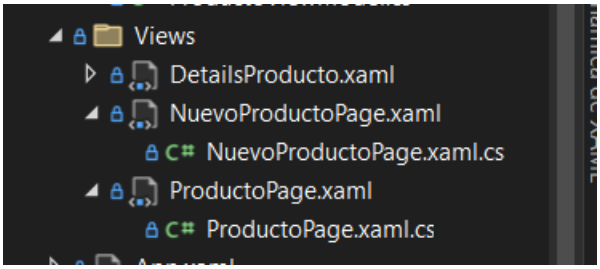
La pantalla que se va a crear es la siguiente:

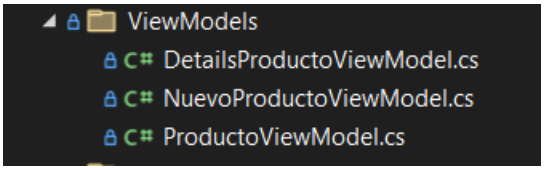

Android Emulator - pixel_5_-_api_34:5554
1:27
NuevoProductoPage
Nombre
Descripcion
Cantidad
Guardar Proveedor



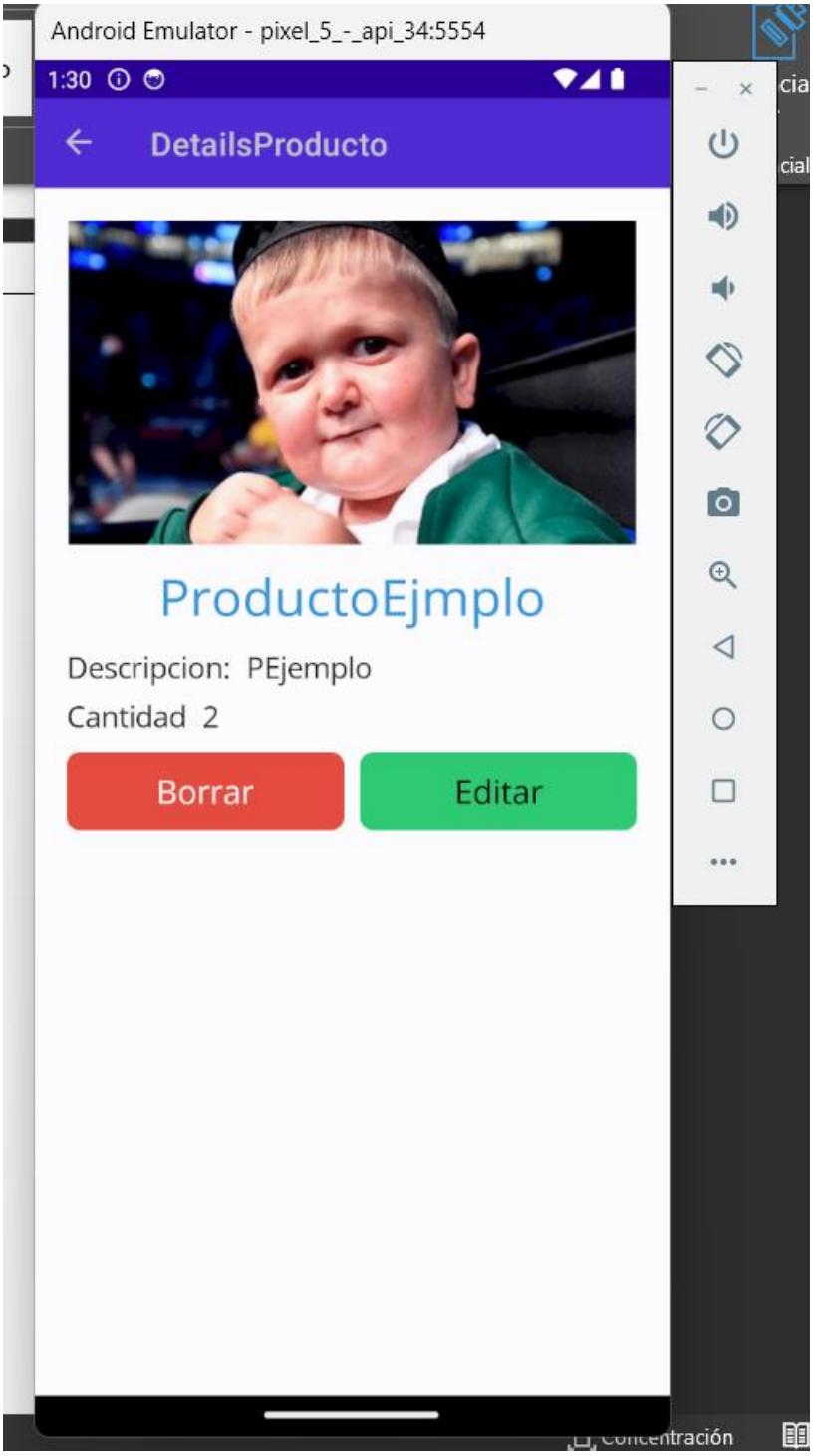


Detalle Producto

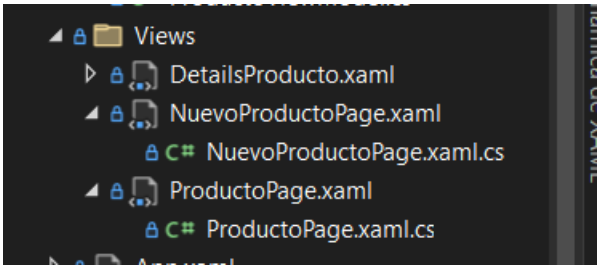


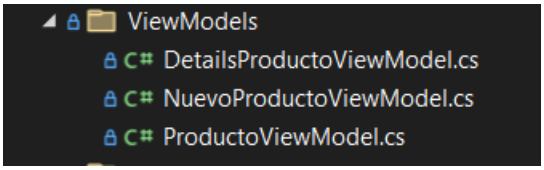


La pantalla que se va a crear es la siguiente:

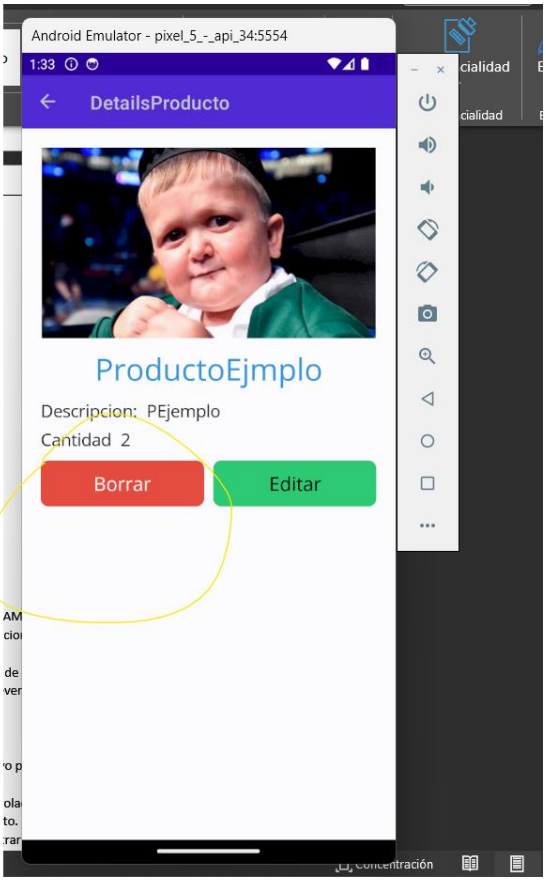


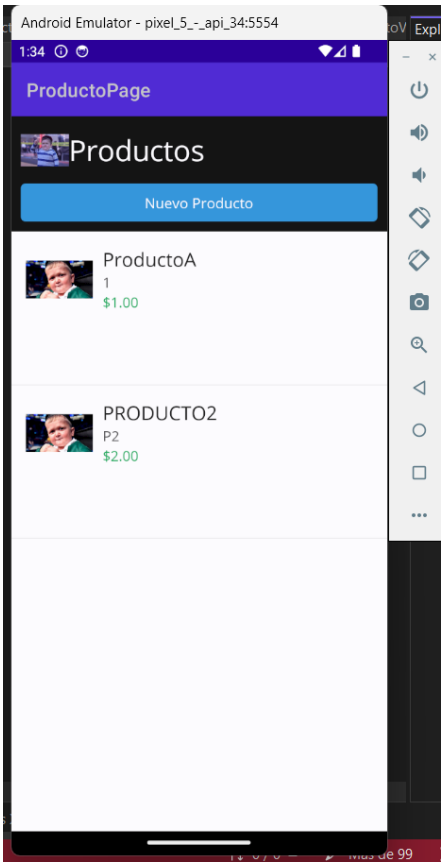
Borrar Producto



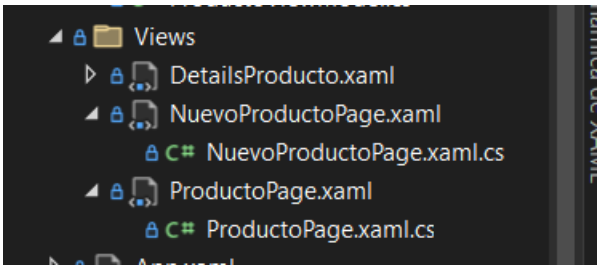


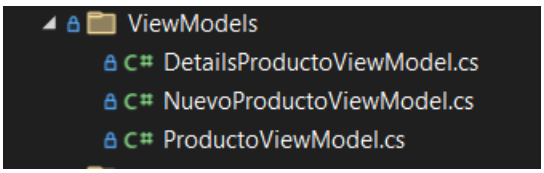
La pantalla que se va a crear es la siguiente:



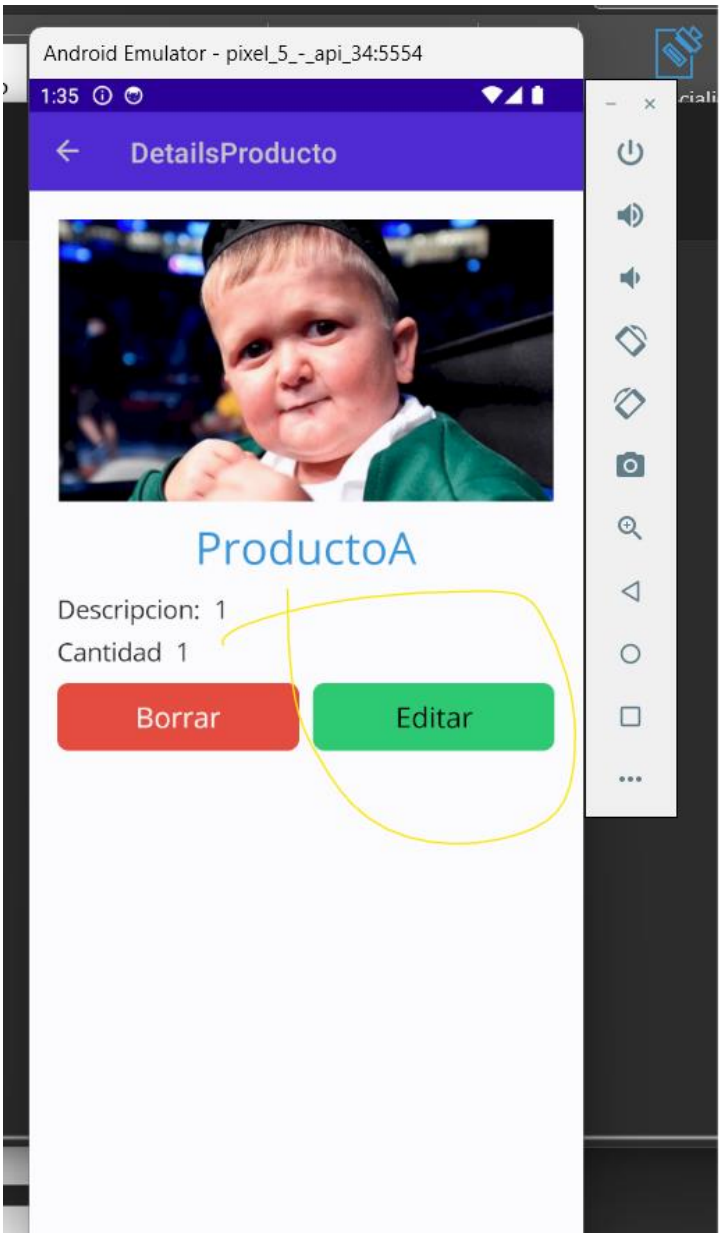


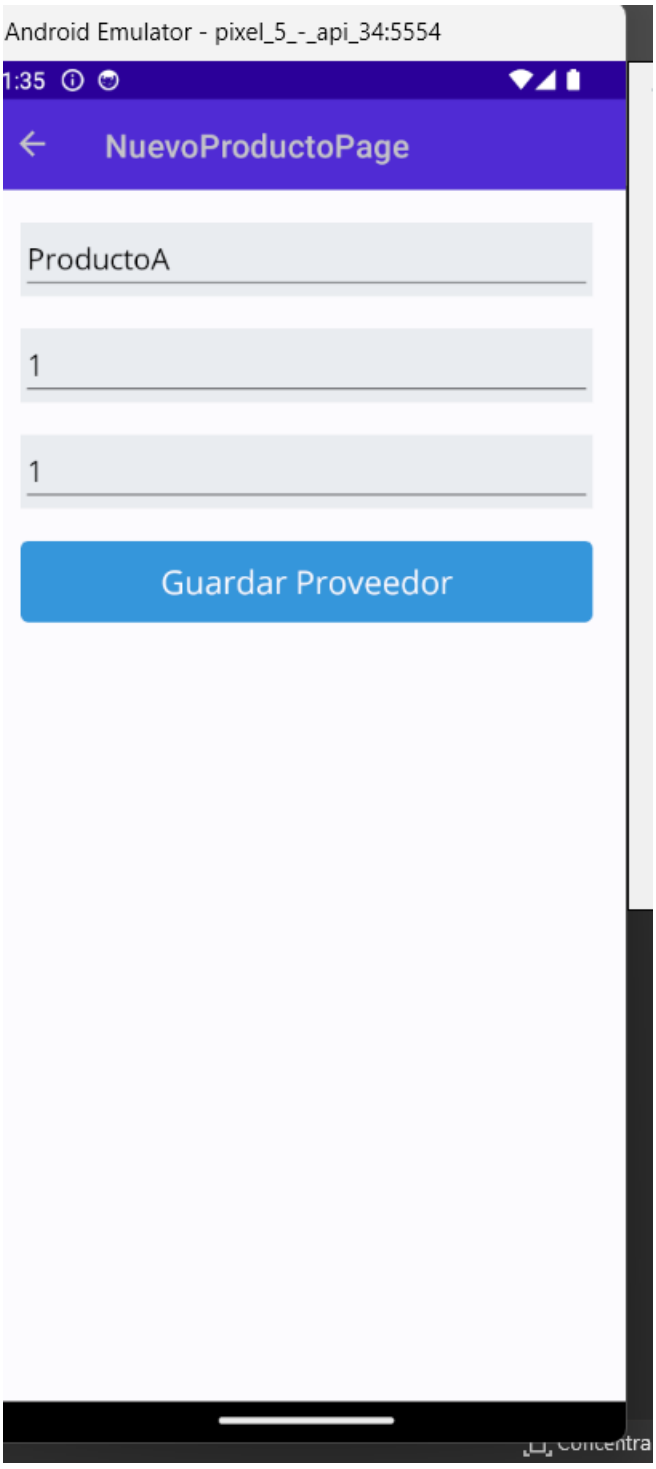
Editar Productos

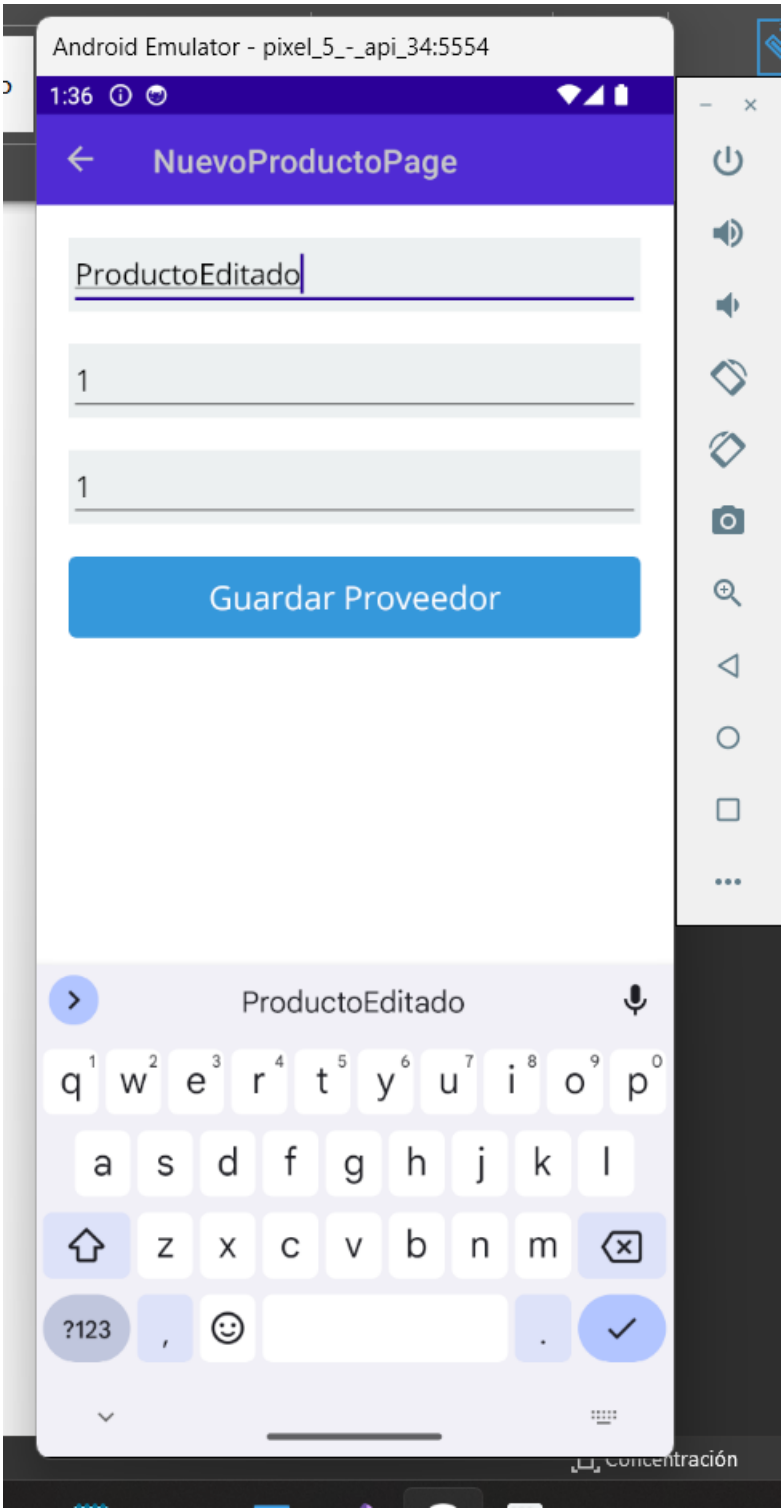


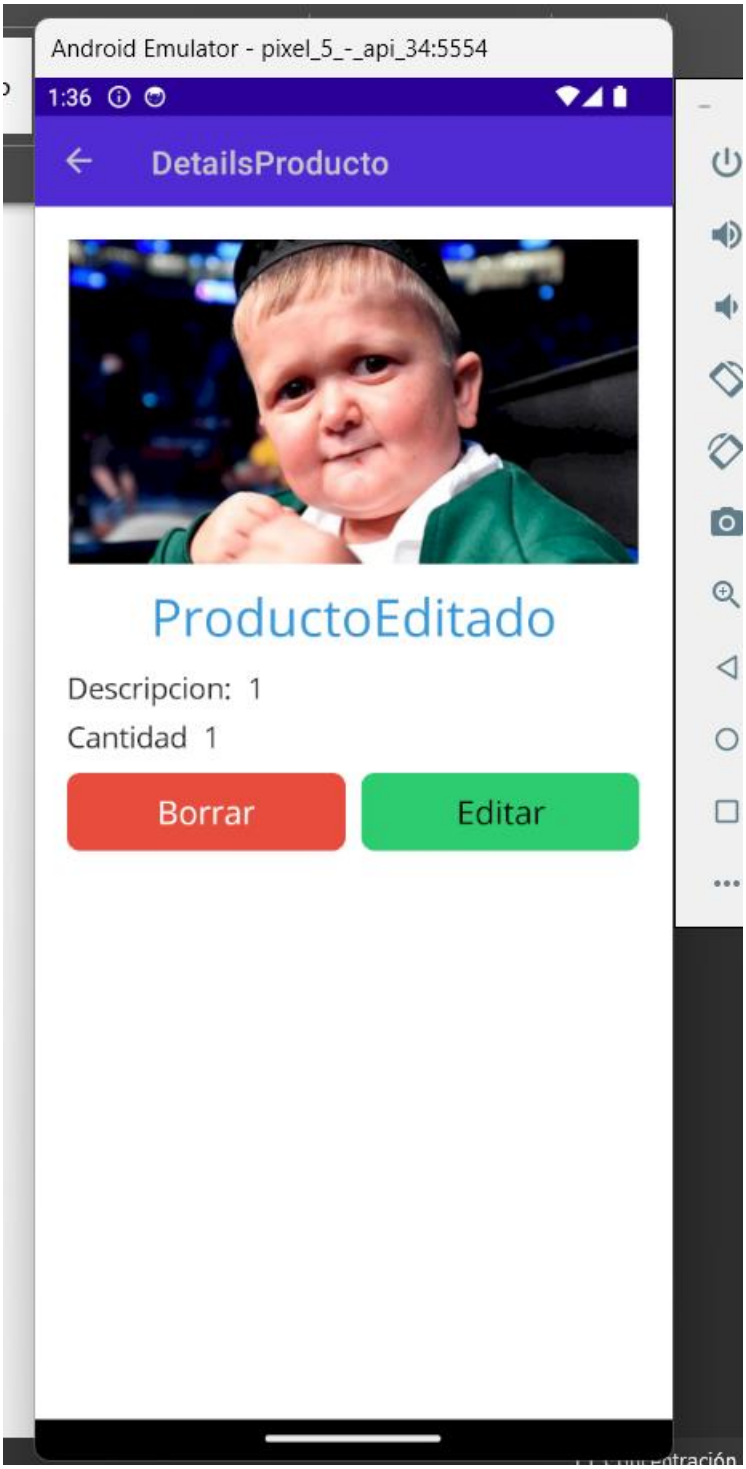


La pantalla que se va a crear es la siguiente:


Android Emulator - pixel_5_-_api_34:5554
1:35
DetailsProducto
ProductoA
Descripcion: 1
Cantidad 1
Borrar
Editar


Android Emulator - pixel_5_-_api_34:5554
1:35
NuevoProductoPage
ProductoA
1
1
Guardar Proveedor


Android Emulator - pixel_5_-_api_34:5554
1:36
NuevoProductoPage
ProductoEditado
1
1
Guardar Proveedor
ProductoEditado
Concentración



Ventajas del Patrón MVVM:

- Separación de Responsabilidades: MVVM separa de manera clara las responsabilidades de la lógica de presentación, la lógica empresarial y los datos, lo que facilita el mantenimiento y la escalabilidad del código.

- Facilita las Pruebas Unitarias: El patrón MVVM permite realizar pruebas unitarias más efectivas, ya que la lógica empresarial se encuentra en el ViewModel y no depende directamente de la interfaz de usuario.

Conclusiones :

- Adopción del Patrón MVVM: La aplicación ha adoptado el patrón de diseño MVVM (Model-View-ViewModel) para separar de manera efectiva la lógica de presentación de la interfaz de usuario (View) de la lógica empresarial (ViewModel) y de los datos (Model). Esto mejora la modularidad y la capacidad de mantenimiento del código.

- Uso de Bibliotecas y Paquetes de Apoyo: Se ha utilizado la biblioteca "PropertyChanged" para implementar la notificación de cambios en las propiedades y mejorar la comunicación entre la Vista y el ViewModel. Esto simplifica la actualización de la interfaz de usuario cuando cambian los datos en el ViewModel.

- Reutilización de Patrones y Estructuras: Se ha replicado el patrón MVVM para múltiples entidades en la aplicación, como "Marca," "Producto," y "Proovedor." Esto demuestra la capacidad de reutilización y escalabilidad del patrón MVVM en la aplicación.
- **6.Link GITHUB**
  https://github.com/EstebanEr-03/ProductoMvvmSqlLiteFinal.git