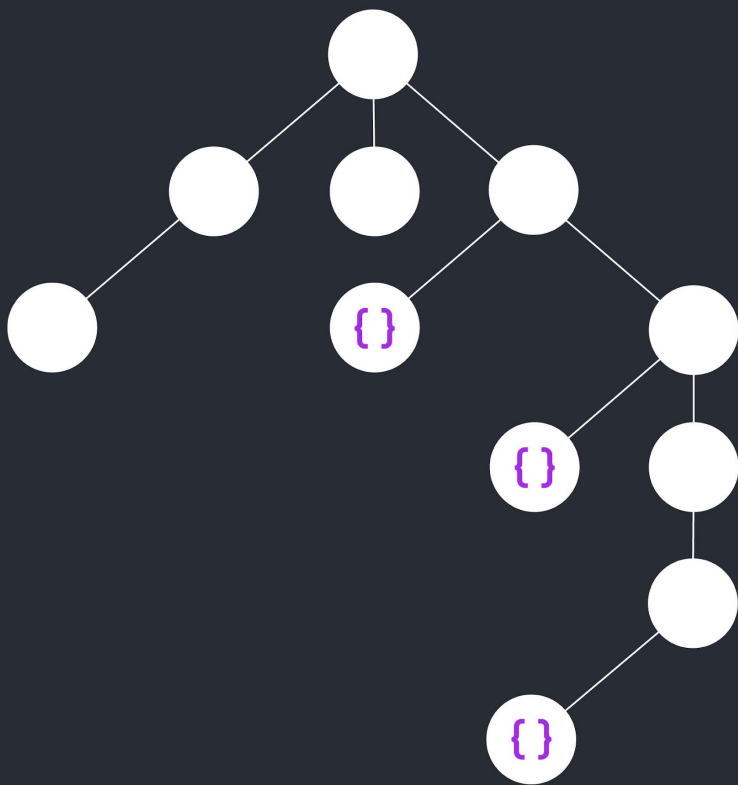


Context

Estado global

DEV.F
DESARROLLAMOS(PERSONAS);

dev

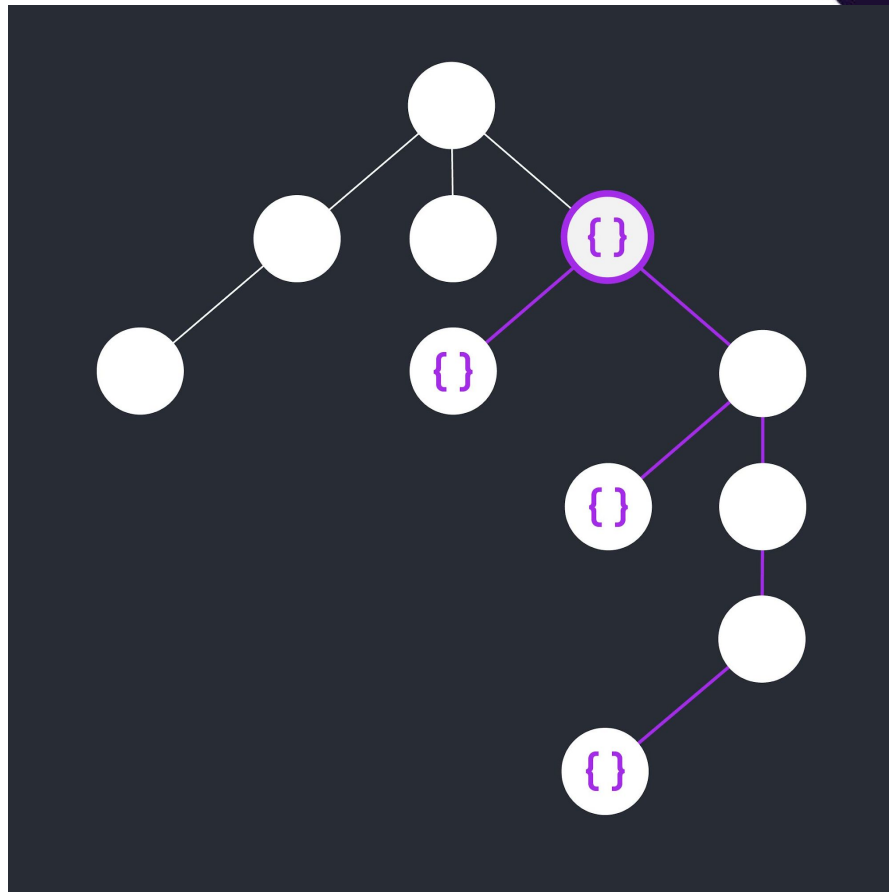


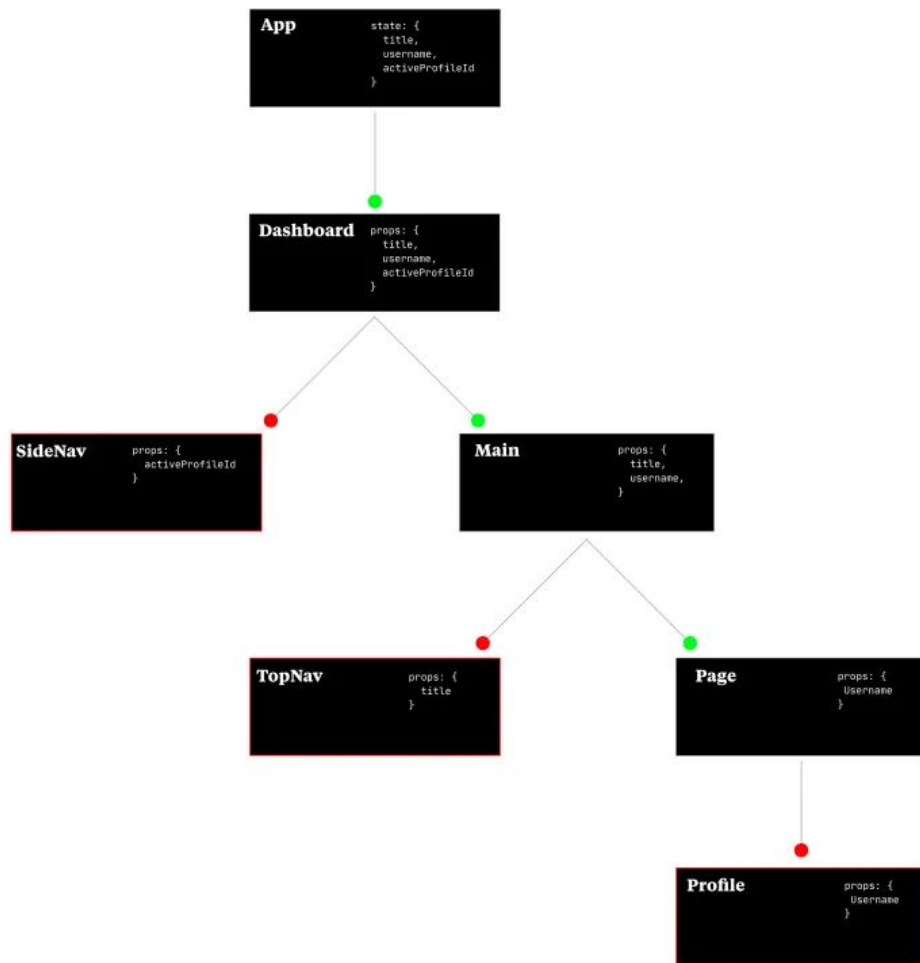
En React todo es un árbol, el componente con mayor poder jerárquico comúnmente maneja la lógica y comparte por props a los demás componentes hijos, esto permite tener:

- Componentes con lógica
- Componente visuales.

Problemática

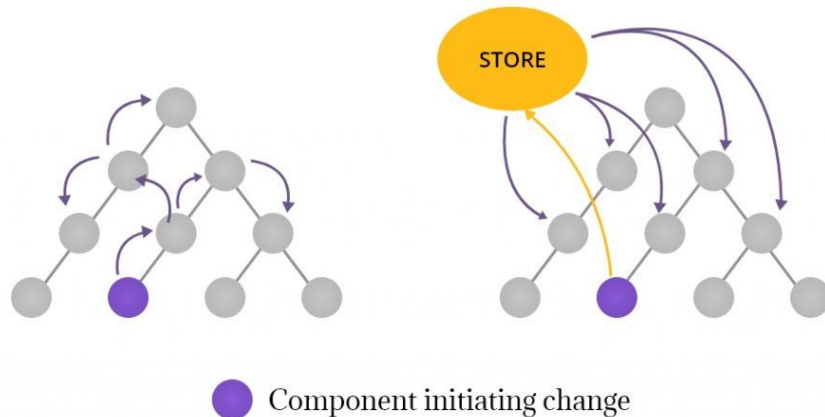
A veces el componente lógico tiene la necesidad de pasar por props los elementos a sus componentes hijos, pero a veces hay demasiada profundidad de componentes o a veces se necesita subir la lógica un nivel arriba para compartir con más componentes.

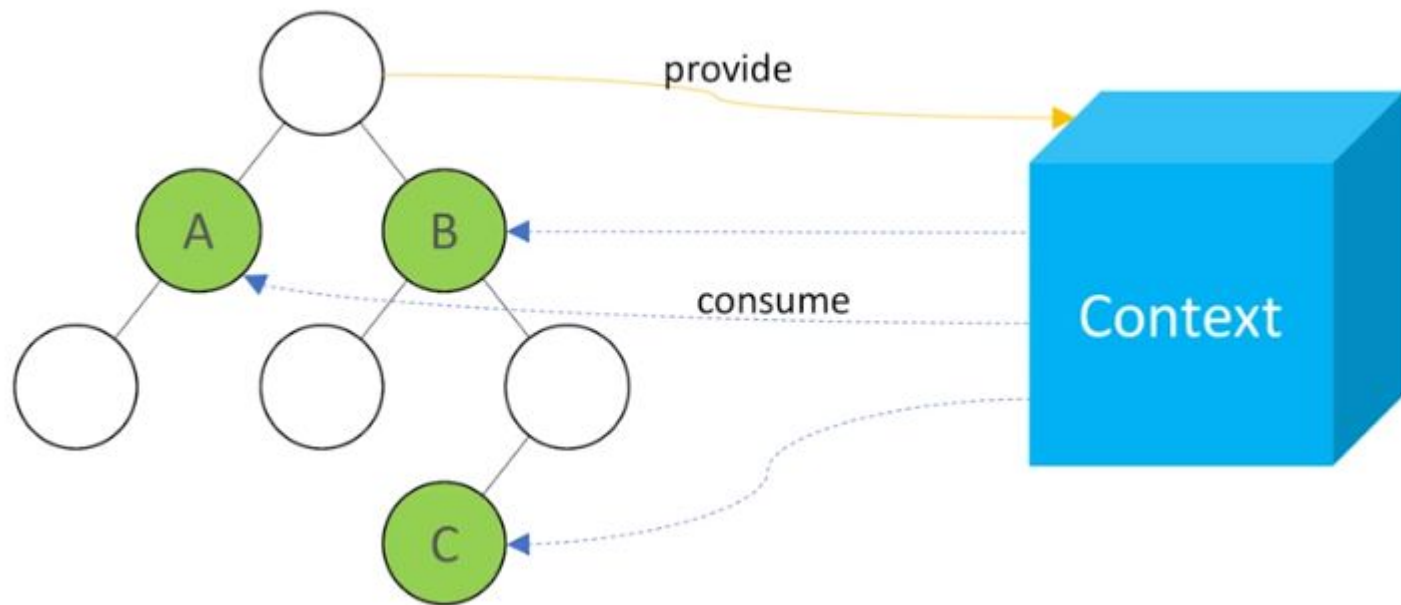




Estado Global

- Un estado global permite tener un estado donde cualquier componente puede acceder.
- No solo puede compartir información, si no funcionalidades completas.
- No afectas a componentes pasandole props que no necesitan.



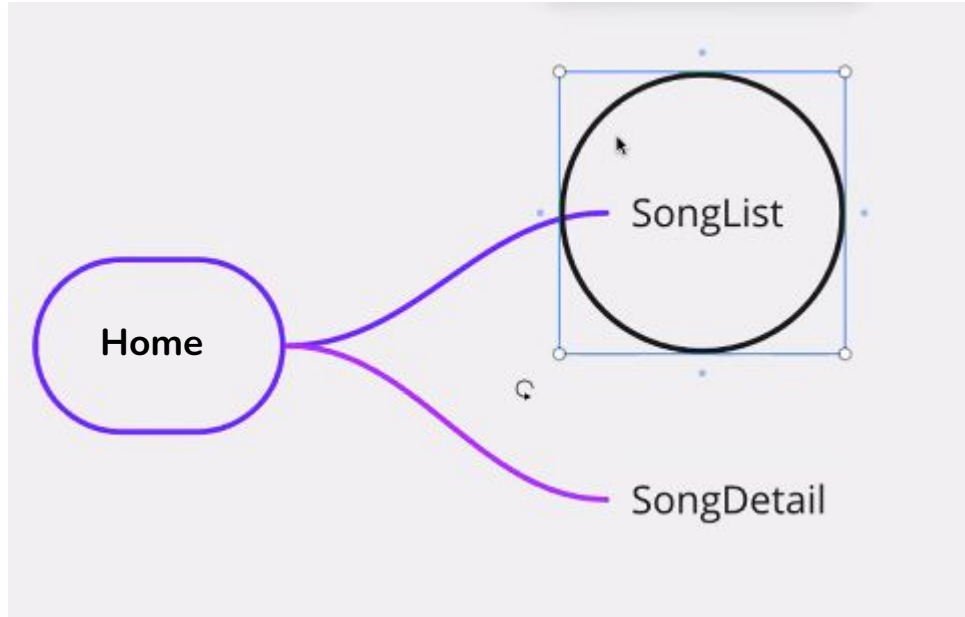


The logo consists of the text 'DEV.F.' in a bold, white, sans-serif font. The 'F' is stylized with a grid of small squares at its top right corner.

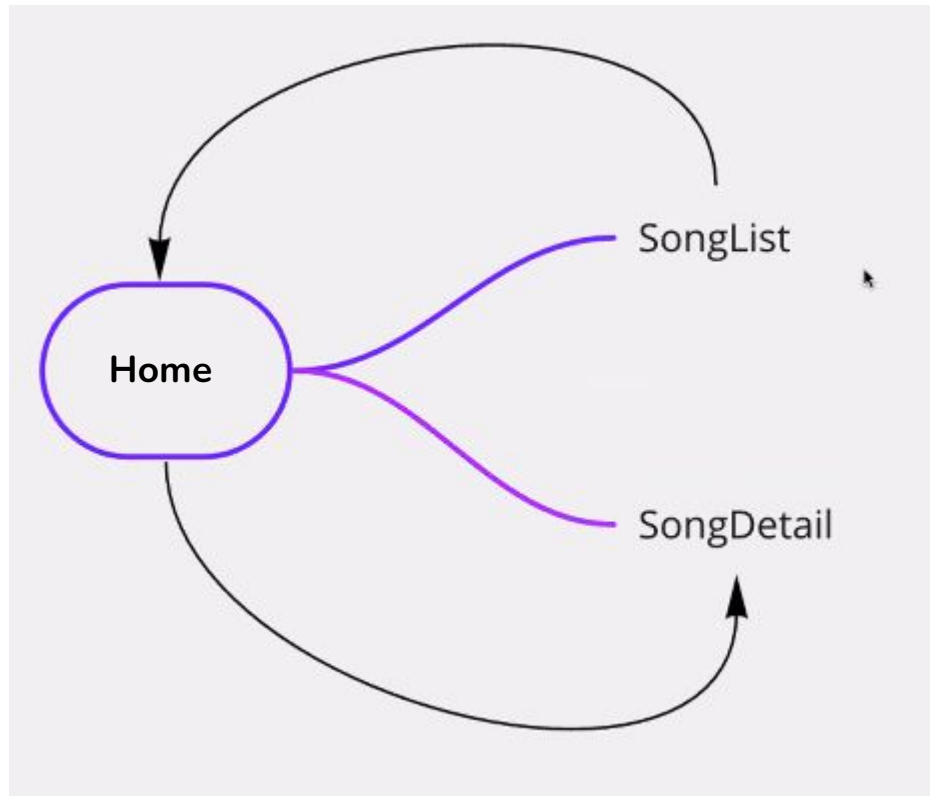
DEV.F.

*DISCUSIÓN EN
ARQUITECTURA*

dev



Si donde mandamos a llamar la información de la API es en **SongList**, entonces el otro componente **SongDetail** no tiene la forma de tener acceso a esa información.



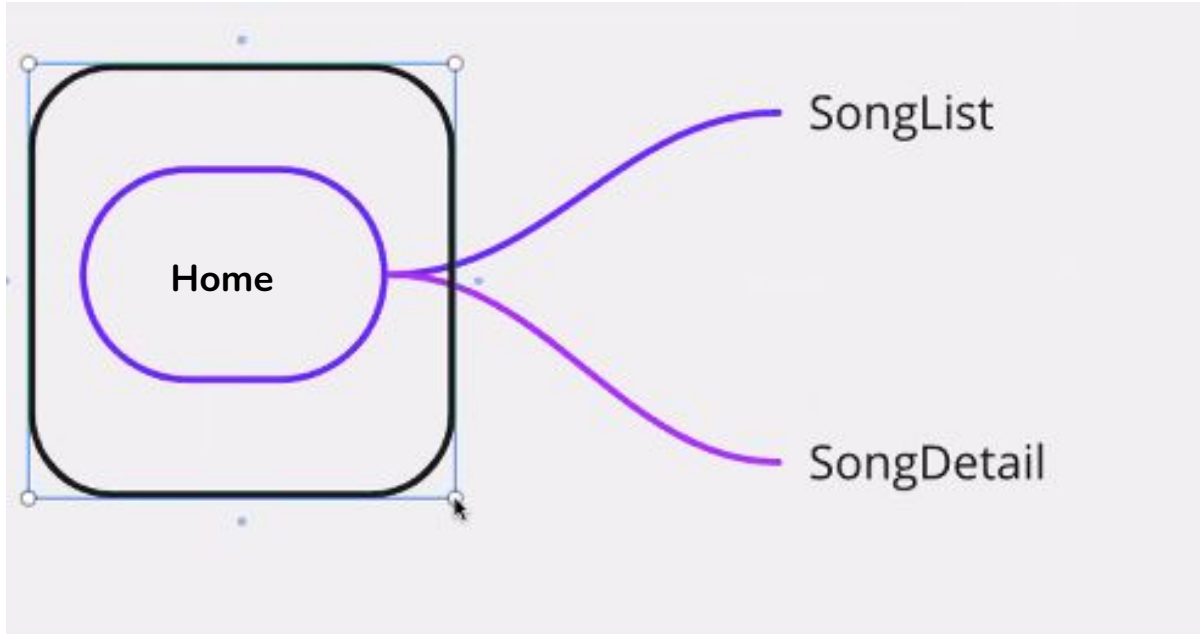
Una posible solución es: Que **SongList** envíe la información de la canción seleccionada a Home y de ahí a **SongDetail**, pero esto no es viable por que la información da más vueltas y complica la lógica.

The logo consists of the text 'DEV.F.' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its top right corner. The logo is centered within a dark blue diamond shape.

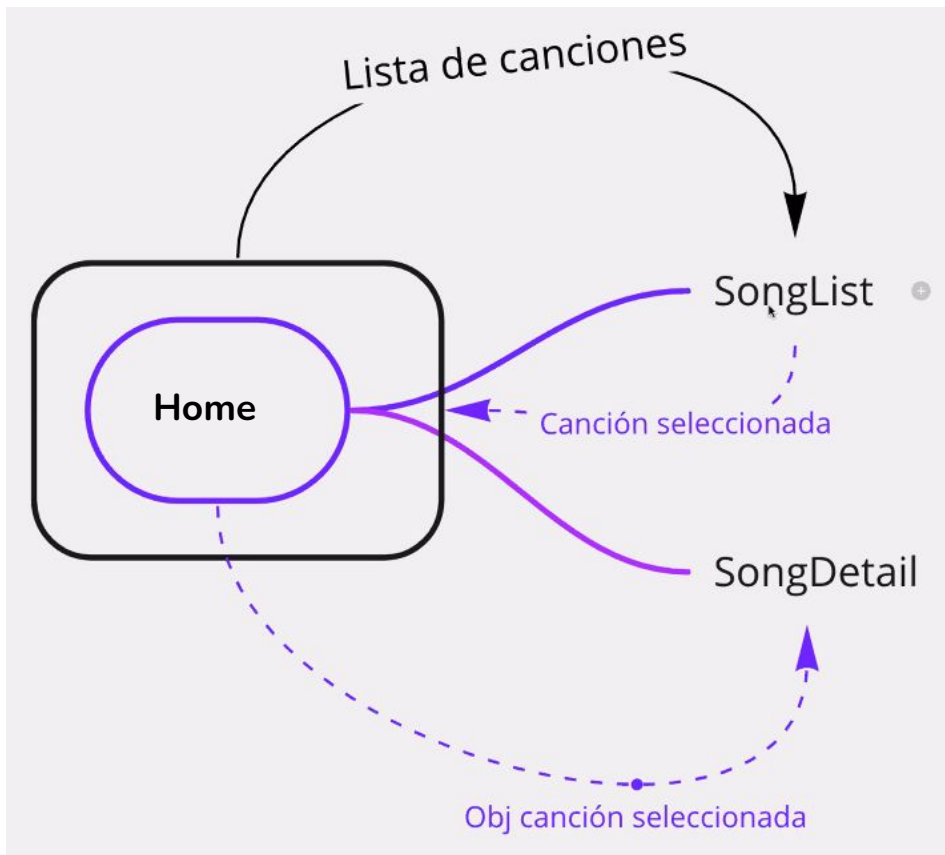
DEV.F.

*NO SE PUEDEN COMPARTIR
DATOS ENTRE COMPONENTES
DE LA MISMA JERARQUÍA.*

*SOLO DE PADRE A HIJO Y
VICEVERSA*



La solución más viable es mover la lógica de quien pide la información al padre de ambos: **Home**



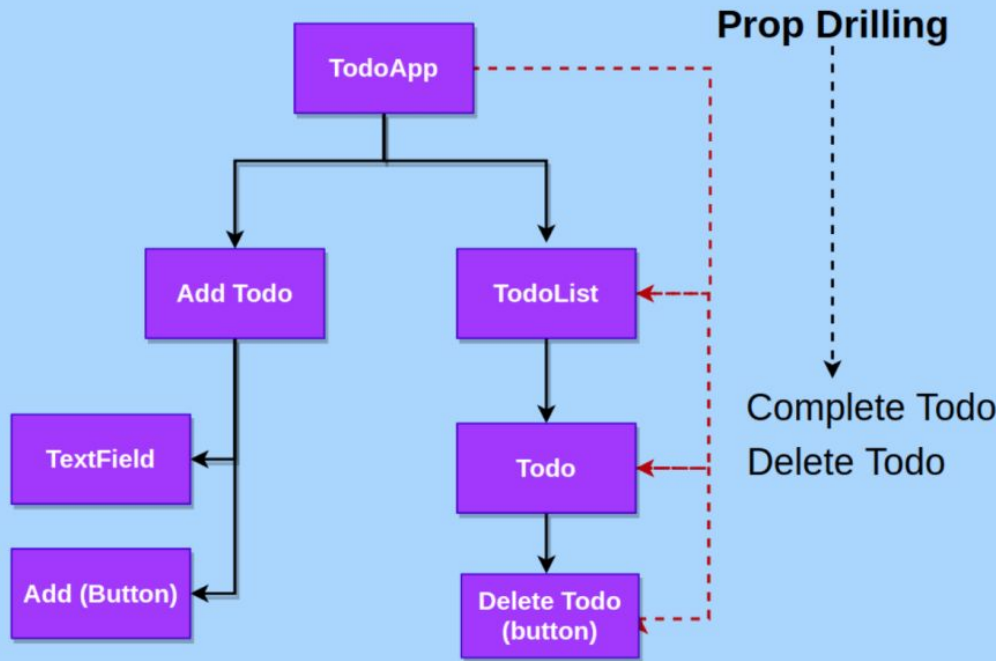
Al mover la lógica a **Home**, se manejaría con props:

1. **Home** obtiene la información de la API.
2. **SongList** recibe la lista de canciones de **Home**.
3. **SongList** envía de regreso a **Home** la canción seleccionada.
4. **Home** notifica a **SongDetail** que una canción fue seleccionada.
5. **SongDetail** pinta esta información.

Este enfoque hace mucho ir y venir, el problema es que existe mucha circulación de información y **Home** realmente no necesita conocer toda esta información que manejan sus hijos.

How to handle Prop Drilling?

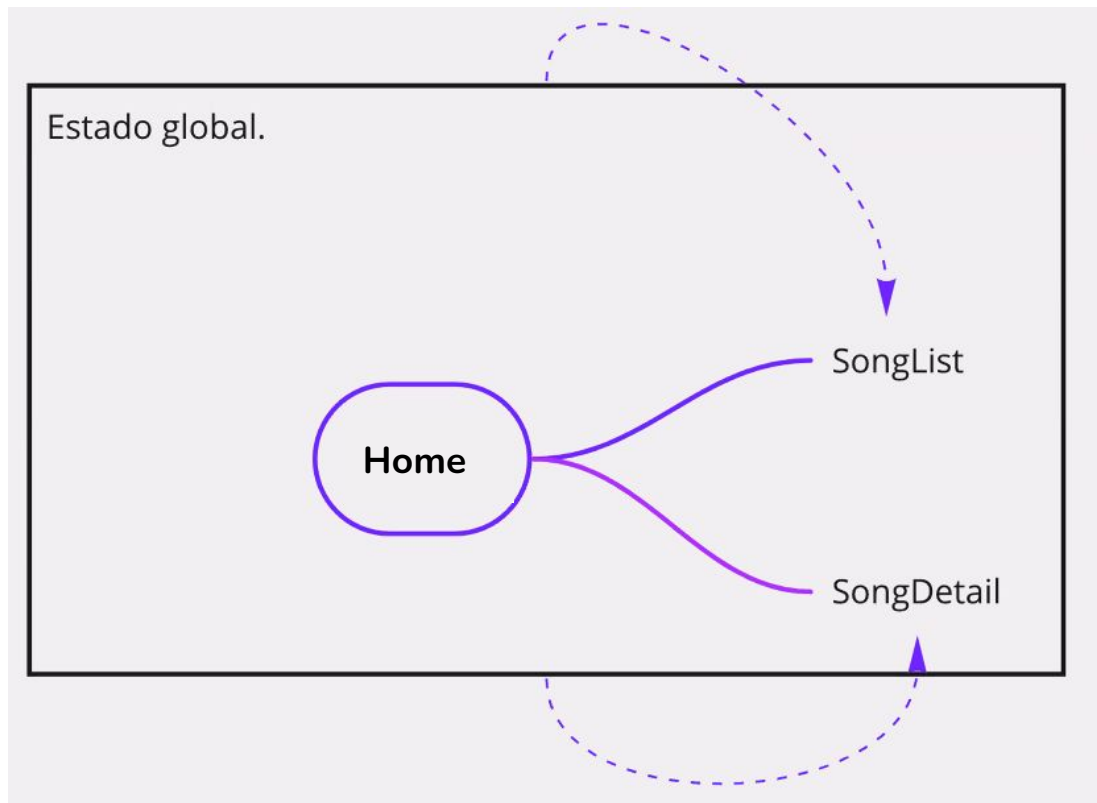
Problema: Prop Drilling



La Solución: Context

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Con ayuda de **Context**, podemos crear un estado global que tenga de habilidad de compartir las listas de canciones y la canción seleccionada.