

# Resumen: Modelos y persistencia

Reflexiona sobre cómo creaste un modelo de datos de Friend, potenciaste una interfaz de usuario usando ese modelo y luego usaste SwiftData para guardar esos datos en los lanzamientos.

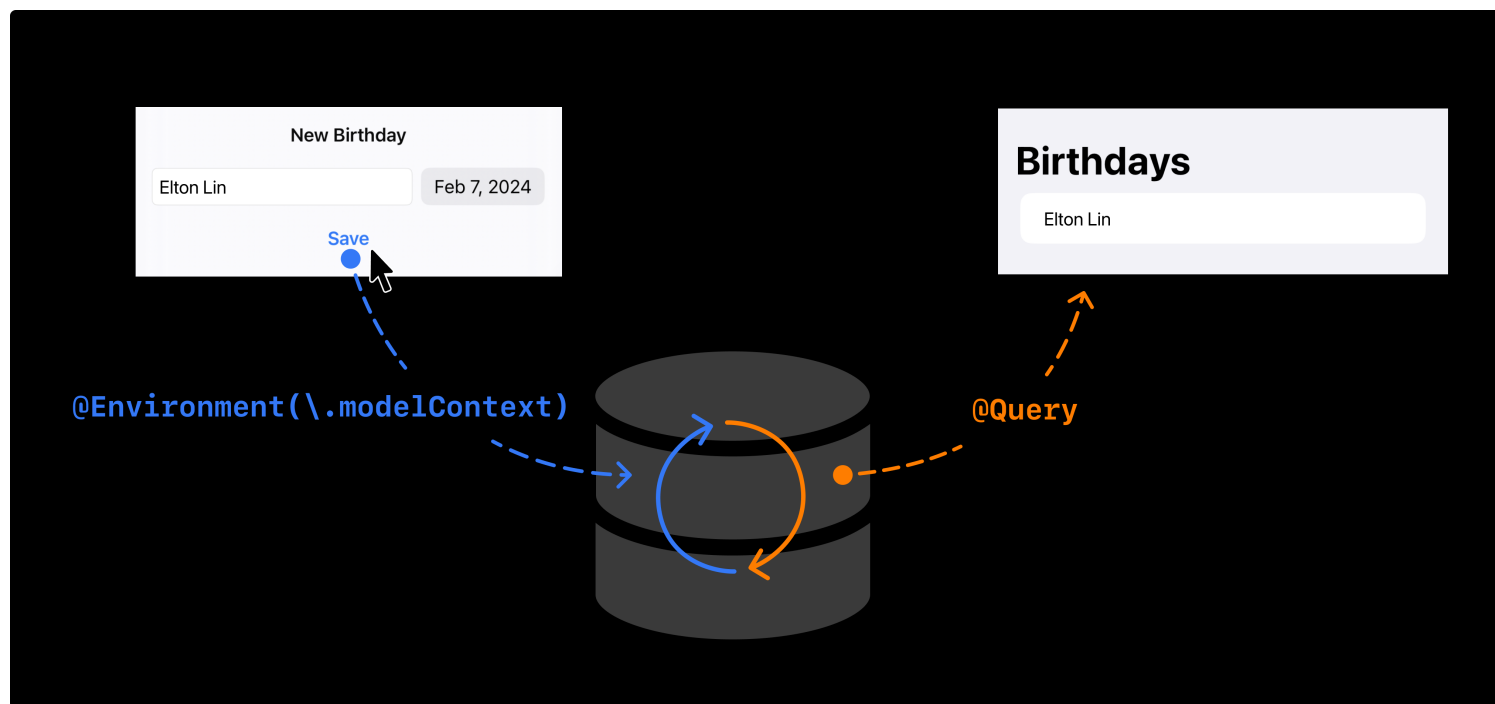
## Reflexionar sobre los conceptos

### Modelos

Los modelos representan el estado de una aplicación fuera de la jerarquía de vistas, incluso en todos los lanzamientos. Ahora tienes otra poderosa herramienta para ensamblar aplicaciones complejas. Casi todas las aplicaciones usan la persistencia de alguna forma, y conoces una forma de hacer que eso suceda.

### Frameworks

The SwiftData framework provides tools to connect SwiftUI to storage.



SwiftData no será el único marco que importe en sus proyectos. Hay una variedad de marcos disponibles para admitir una amplia gama de funcionalidades de la aplicación. Aquí hay algunos ejemplos:

- Swift Charts proporciona una potente visualización de datos
- Contactos ofrece acceso a los contactos
- PhotoKit facilita el acceso a las fotos y vídeos almacenados

### Temas y habilidades cubiertos

- \* Representando conceptos del mundo real como modelos de datos
- \* Uso de Date, DatePicker y Calendar para trabajar con las fechas
- \* Posicionar el contenido en la pantalla usando .safeAreaInset
- \* Formato de fechas en vistas de texto
- \* Clases de SwiftData @Model
- \* Contexto de SwiftData
- \* Uso de @Query para rellenar una interfaz
- \* Usar @Environment para obtener valores importantes de las vistas de los padres

## Continuar practicando

### Amplía tu aplicación

1. Ordena la lista de cumpleaños por nombre en lugar de por cumpleaños.
2. Añade una propiedad de notas a Friend para planificar cómo celebrarás el cumpleaños de un amigo.

### Ponte a prueba

Vuelva a visitar Pick-a-Pal y reemplace los nombres: [String] con modelos Pal con SwiftData.

### Más información

Para obtener más información sobre modelos y clases, lea Estructuras y clases. Para obtener más información sobre los inicializadores, lea Inicialización.

## ¿Qué sigue?

SwiftData puede hacer mucho más que guardar y recuperar un solo tipo de modelo. En el siguiente tutorial, aprenderás a crear y gestionar datos de muestra, a mostrar y editar modelos en detalle, y a crear vistas personalizadas flexibles para los modelos.

## Crear, actualizar y eliminar datos

Cuando basas una aplicación en un almacén de datos común, debes dar a las personas una forma de crear nuevos elementos, actualizar los existentes y eliminar elementos que ya no necesitan.

