



Universidad
Tecmilenio®



Desarrollo de aplicaciones en plataforma iOS

Construcción de
prototipos





Mientras pienses en cómo crear tu app, deberás tener en cuenta el diseño, y necesitarás habilidades nuevas que usarás mucho antes de escribir el código.



Ícono

Medio más poderoso para convencer a un usuario que obtenga tu app.

Pantallas de inicio

Cuando se carga una app, se mueve a esta pantalla mientras se prepara para la pantalla de destino principal.

Color

El uso sutil constante de color es una forma de comunicar tu marca y personalidad.

Modo oscuro

Apariencia oscura de todo el sistema.



Imágenes e iconos

El buen uso de imágenes e iconos personalizados agrega claridad, atractivo visual y singularidad a tu app.

Animación

Son una forma poderosa de conectar a los usuarios con contenido y crear una personalidad única.

Tipografía

Puedes elegir entre cientos de tipos de letra y estilos útiles.



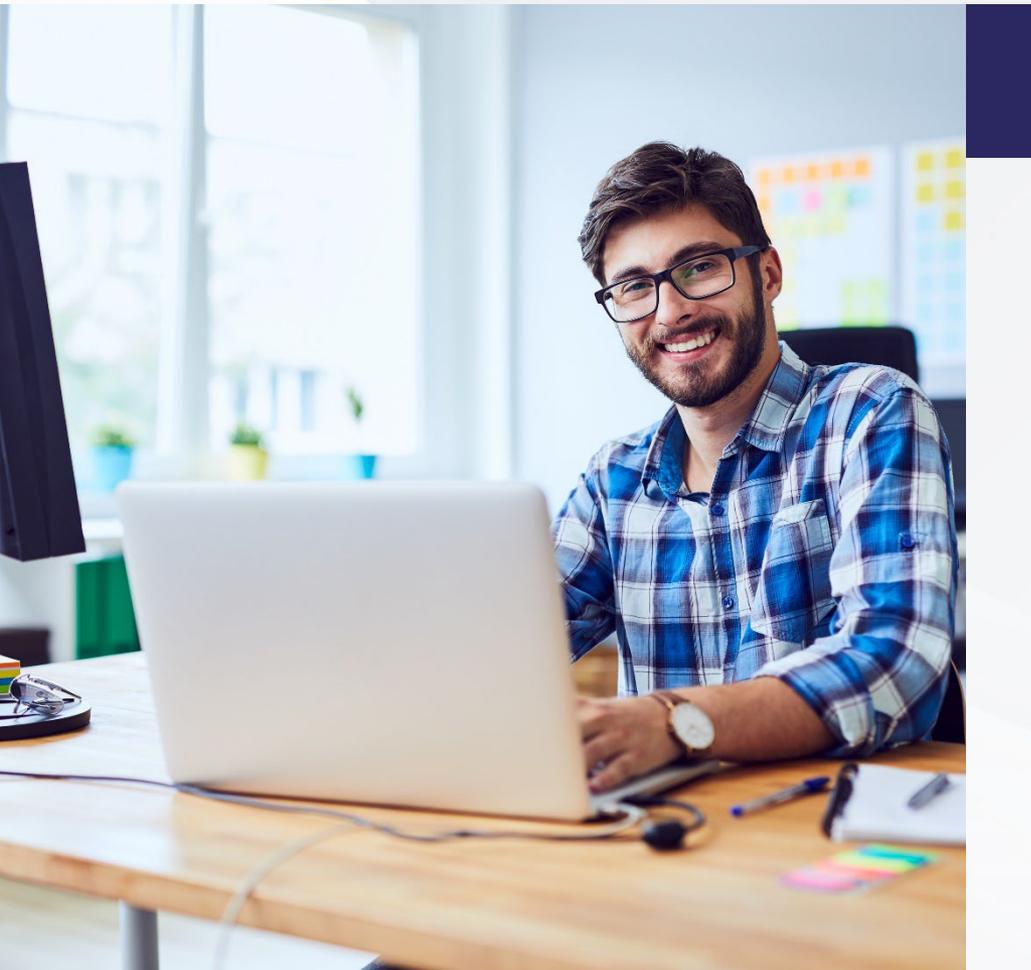


Fuente: Apple Education. (2021). *Develop in Swift Fundamentals*. EE. UU: Apple Inc. – Education.
Recuperado de <https://books.apple.com/us/book/develop-in-swift-fundamentals/id1556365994>



- ¿Cuáles son los pasos del ciclo del diseño?
- Menciona 3 elementos que ayudan a mejorar el aspecto visual de tu aplicación, así como sus recomendaciones.





A esta altura, ya debes confiar en tus habilidades para crear apps sencillas con pocas instrucciones. Recuerda que el proceso de diseño es de suma importancia previo a ponerte a codificar tu aplicación, de tal forma que obtengas el producto que tus usuarios están buscando.



Desarrollo de aplicaciones en plataforma iOS

Patrones de diseño





Los controladores de la barra de pestañas son clave para navegar entre los controles de vista, lo que le permite empaquetar cómodamente más funciones en una sola aplicación.



Controlador de
barra de pestañas

Permite organizar la
app según distintos
modos o secciones

Estados de ciclos
de vistas:

- No cargada
- Aparece
- Aparecida
- Desaparece
- Desaparecida



Modelos: Las pantallas de entrada de información están atadas a un modelo de datos

Pantallas de entrada: Son una forma de controlar información ingresada (formularios)

Recolectando strings: debes usar campos de texto

Recolectando fechas: utilizar un recolector de fechas (poner 2 celdas)

Recolectando números: sliders, steppers o text fields

Recolectando entradas binarias: Usar un Switch





- Contesta brevemente las siguientes preguntas:
- ¿Qué propiedad modificarías si quieres agregar una insignia a una pestaña?
- ¿Qué valor asignarías a la respuesta anterior para ocultar una insignia?
- ¿Cuándo se debe usar un controlador de barra de pestañas?
- ¿De los métodos del ciclo de vida de un controlador de vista, cual se ejecuta primero?





Como puedes ver, hay muchos usos para los métodos de ciclo de vida del controlador de vista. Cada uno es como una "notificación" que le dice a tu código que el evento de visualización ha tenido lugar. Al usar esta guía, tú como desarrollador descubrirás la mejor manera de aprovechar cada uno de los métodos para la tarea particular en cuestión.

Por otro lado, la entrada del usuario puede ser muy difícil de manejar. Aprender a recopilar y administrar adecuadamente esta información es una parte increíblemente importante para convertirte en un desarrollador de aplicaciones. A medida que se te ocurran tus propias ideas para la aplicación, asegúrate de considerar cómo puedes usar lo que aprendiste en esta lección para manejar mejor los comentarios de los usuarios.



Desarrollo de aplicaciones en plataforma iOS

Vista de tablas





MVC, que significa Modelo-Vista-Controlador. MVC te ayudará a diseñar los archivos en tu aplicación, así como las interacciones y relaciones entre diferentes tipos e instancias.



MVC

Ayuda a la arquitectura de los archivos, así como la interacción y relación de diferentes tipos de instancias.

Capas MVC

- Model
- View
- Controller

Anatomía de una vista de tablas

Es una instancia de la clase UITableView

Controladores de vistas de tablas

Es una subclase que maneja una instancia de una vista de tablas simple.



Estilos de vista de tablas

- Plano
- Agrupado
- Recuadro agrupado

Celdas de una vista de tabla

- Cada fila se representa con una celda



Vistas de desplazamiento

Permite mostrar contenido que no quepa en toda la pantalla, al igual que su manipulación.





- Contesta las siguientes preguntas:
- ¿Un view controller es un modelo, vista o un objeto de control?
- ¿Un model controller es un modelo, una vista o un objeto de control?
- ¿Qué métodos de origen de datos se requieren para las vistas de tabla?
- ¿Puedes unir una vista de tabla sin un controlador de vista de tabla?





Con tus nuevas habilidades podrás construir con confianza y eficiencia una pantalla personalizada para presentar cualquier objeto modelo que use tu aplicación y responder a las interacciones del usuario con su vista de tabla.

Recuerda que `UIScrollView` es la clase principal de varias otras clases en `UIKit`, incluidas `UITableView` y `UICollectionView`. Estas dos clases prominentes heredan toda la funcionalidad de las clases `UIScrollView`. A medida que avances en el desarrollo de tu app, querrás tener en cuenta esta relación, ya que podrás usar todas las herramientas que aprendiste en esta lección cuando uses vistas de tabla o vistas de colección.



Desarrollo de aplicaciones en plataforma iOS

Vista de tablas II



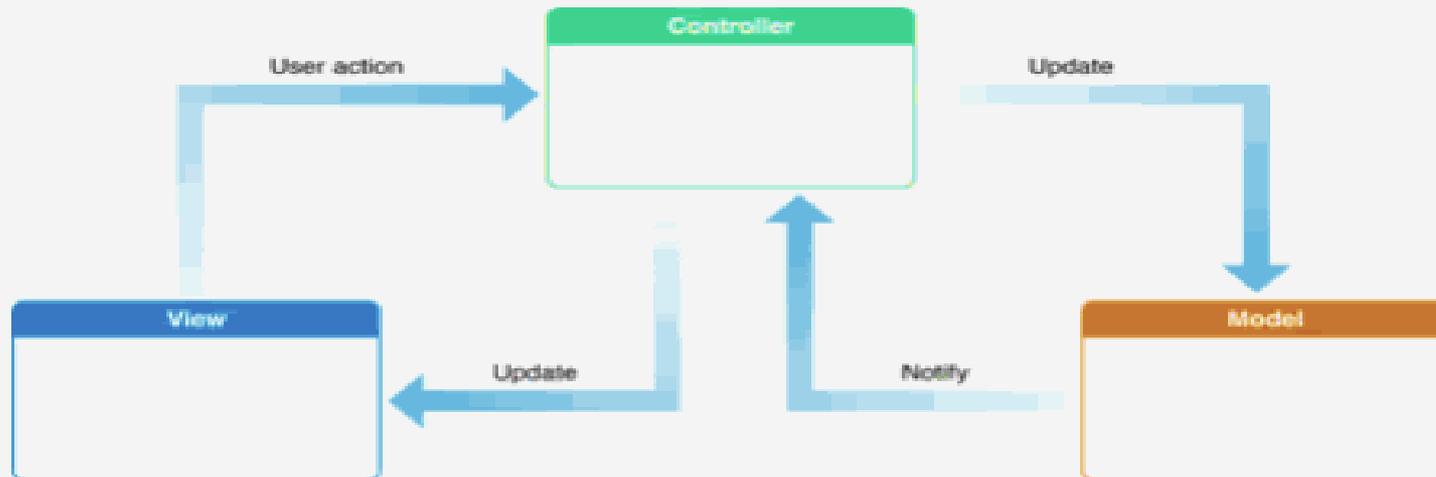


El sistema operativo se ejecuta en dispositivos potentes que pueden manejar tareas informáticas sofisticadas, lo que permite a los usuarios trabajar con grandes conjuntos de datos. Como programador, ¿cómo puede mostrar datos para que sean elegantes de ver y eficientes de manipular?

En este tema, aprenderás por qué las vistas de tabla son tan populares entre los desarrolladores de iOS. Y también cómo personalizarás aún más tus vistas de tabla con celdas personalizadas, cómo insertar y eliminar celdas dinámicamente, cómo usar vistas de tabla estáticas para presentar formularios u otras listas que no cambian, y cómo permitir que el usuario deslice el dedo para eliminar celdas.

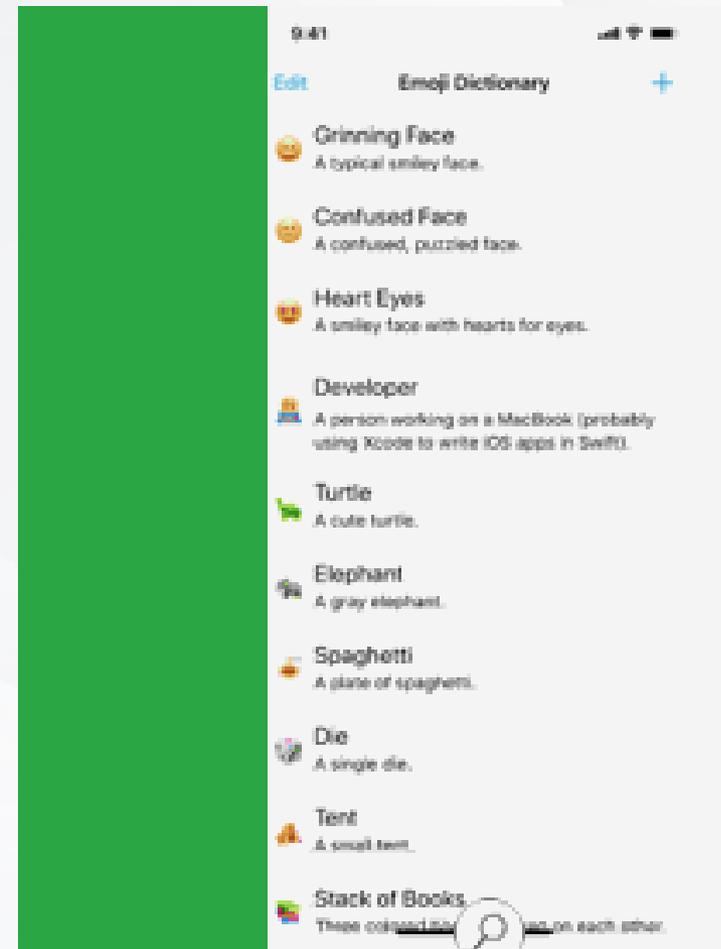


Como programador, deberá diseñar sus aplicaciones para que guarden los datos del usuario entre lanzamientos. Imagínese si un usuario escribiera una nota en la aplicación Notas y desapareciera cuando la reiniciarán. Ya sea diez segundos o diez meses después, los usuarios esperan que todos sus datos sigan allí.



Arquitectura MVC. Fuente:
Capítulo 1. Subtema 1.5. Apple Education. (2021). Develop in Swift Data Collections. EE. UU: Apple Inc. – Education.

A lo largo de esta sección, se basará en el proyecto EmojiDictionary que configuró en la lección anterior de vista de tabla. En esa lección, usó los estilos de celda de vista de tabla proporcionados en el SDK de iOS. Pero, ¿y si quisiera una mayor personalización? Por ejemplo, las celdas de la vista de tabla en la aplicación Correo muestran texto de vista previa para que el usuario pueda leer una parte de cada mensaje de un vistazo. En la App Store, las celdas muestran los nombres de las aplicaciones y una imagen de la aplicación, así como un botón que permite a los usuarios descargar la aplicación directamente desde la vista de tabla.



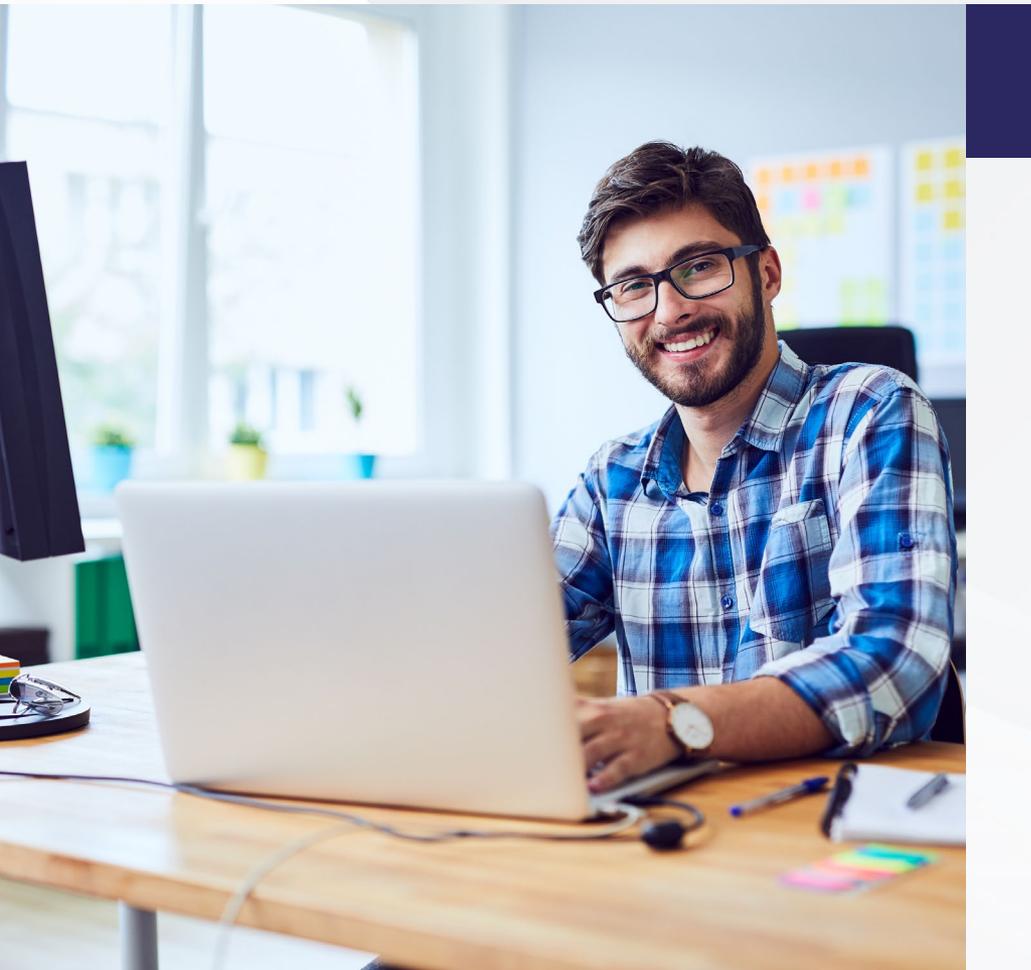


- Mira este video sobre las vistas de tablas, analízalo y posteriormente realiza un resumen de lo aprendido.



Andrés Cruz. (2019, 17 de octubre). *5 TableView y los UserDefaults para guardar la data de los usuarios en Swift* [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=EuCh_M2Afao





En esta lección, aprendiste por qué las vistas de tabla son tan populares entre los desarrolladores de iOS. Además de, cómo personalizar aún más tus vistas de tabla con celdas personalizadas, cómo insertar y eliminar celdas dinámicamente, cómo usar vistas de tabla estáticas para presentar formularios u otras listas que no cambian, y cómo permitir que el usuario deslice el dedo para eliminar celdas.

