



Universidad
Tecmilenio®



Desarrollo de aplicaciones en plataforma iOS

Swift: Fundamentos 1





Dentro del desarrollo en Swift se deben aplicar ciertos criterios a los que se conoce como patrones de diseño, mismos que son recurrentes para la resolución de problemas. Existen varios tipos de patrones.



Condicionales

Comparación de un valor para determinar si corresponde a un criterio.

Ciclos

Realiza de forma repetida una serie de acciones.

Funciones

Crear partes de código externas a la estructura de la App, con la finalidad de simplificar la lógica o de concentrar el desarrollo para una acción específica.

Enumeraciones

Permite definir los tipos de datos basados en el diseño de una estructura.





Después de revisar los contenidos, contesta las siguientes preguntas:

- ¿Qué se utiliza para ejecutar un ciclo sobre una sección de código?
- ¿Qué se utiliza para ejecutar un ciclo a través del rango 0-9 e ignorar el valor del rango?





Al conocer los condicionales podrás aplicar instrucciones para validar si cumple con un criterio de comparación; con el patrón de lógica computacional if/else puedes dirigir el flujo de acciones a realizarse dentro del desarrollo de la App, puedes apoyarte de la instrucción switch que, de forma semejante a if/else, permite mostrar más criterios dentro de un mismo criterio de comparación con el que es posible extender el diseño de tu solución.

Finalmente, podrás crear tus propios tipos de datos (enum) específicos o extender alguno de ellos para establecer una asignación en particular. Todos estos criterios y lógica computacional permitirán que tomes las decisiones de diseño correspondientes para hacer más simple la integración del código con el diseño de solución.



Desarrollo de aplicaciones en plataforma iOS

Swift: Fundamentos 2





La estructura de datos es una forma de agrupar o de encapsular los datos, lo cual ayuda a establecer un acceso para múltiples usos de un tipo de dato.



Clase

Estructura de datos que contiene propiedades y métodos que la hace general para los datos que se agruparán.

Estructura

Se define como un value type, es decir, una estructura que se enfoca en el valor como tipo.

Extensiones

Simplifican algunas de las funciones básicas del lenguaje utilizando un tipo de dato de la librería de Swift o alguna otra extensión creada a la medida.

Opcionales

En algunos casos los tipos de datos que se asignan o se obtienen en un procedimiento pueden ser definidos o nulos,. Para los casos en los que no tenemos control, se utiliza el tipo de dato Optional.





- Revisa la imagen y contesta la siguiente pregunta: ¿Por qué el siguiente código no compila?



```
class científico{  
func performCalculation(){}  
}
```

```
class Physicist: Scientist {  
func performCalculation(){}  
{
```





La correcta definición de la estructura diseña eficientemente los datos para encapsularlos y con ellos tener acceso a valores o métodos: la forma en que se organizan será útil para reutilizar elementos y hacer más eficiente el desarrollo. ¿Cómo es esto posible? Sólo bastará con instanciar en la clase o estructura y utilizarla.

También pudiste conocer sobre el uso de extensions para reutilizar ciertas clases o estructuras que están en la biblioteca de Swift u otras desarrolladas por ti. Las extensiones ayudan a aprovechar los métodos que ahí están declarados y utilizarlos dentro de otra estructura de datos sin necesidad de crear desde el código.



Desarrollo de aplicaciones en plataforma iOS

Swift: Fundamentos 3





La lógica en el código es fundamental al momento de programar tu aplicación, ya que si no entiendes el porqué de la sintaxis de alguna línea, es posible que tu diseño de solución sea desechado.



Guard

Comunica de forma más clara que condiciones del código deben cumplirse.

Para salir de una instrucción guard debes usar un return.

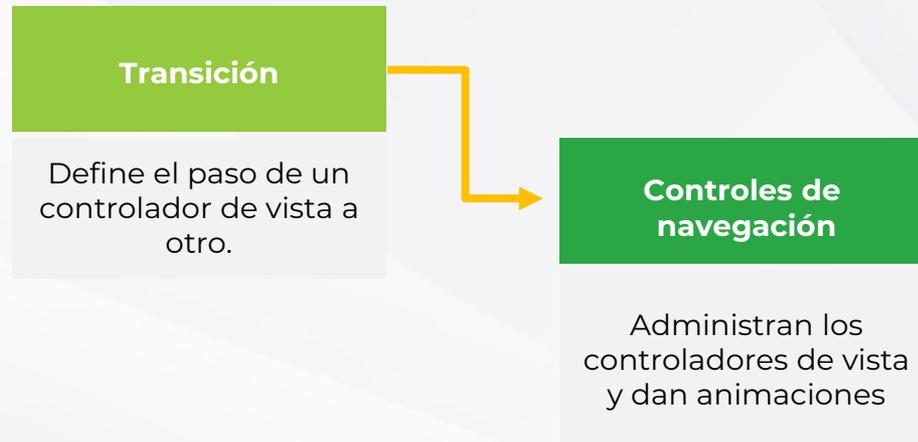
Puedes colocar el código que se espera ejecutar en la parte inferior.

Alcance de variables

Alcance global: accesible desde cualquier parte de la aplicación.

Alcance local: solo es accesible para una función o una parte específica de la aplicación.

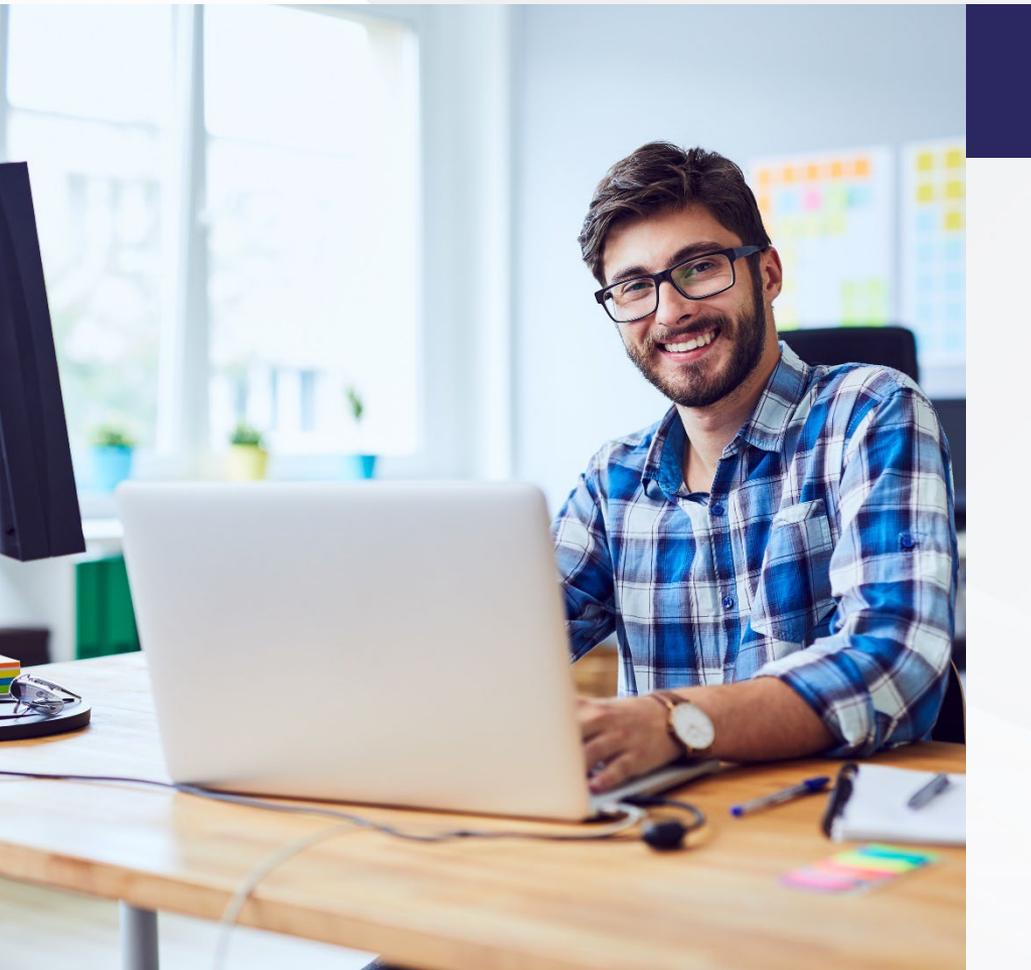






- Revisa los contenidos y contesta lo siguiente:
- ¿Cuál es el propósito de una instrucción guard?
- ¿Cómo declaras una variable global?
- ¿Cómo declaras una variable local?





En el tema observaste que puedes utilizar Guard para lograr tener un mejor flujo de control. También aprendiste acerca de las variables locales y globales, así como su utilidad al momento de generar código. Por último revisaste como se realizan las transiciones para pasar de un controlador de vistas a otro, de tal forma que le de una excelente funcionalidad a tu aplicación.



Desarrollo de aplicaciones en plataforma iOS

Controles, layout y vistas





Anteriormente, ya has trabajado con vistas y controladores comunes. En este tema te enfocarás en crear controladores y dar respuesta a eventos de control.



Botones

Tipo más común de control de entrada de datos.

Interruptores

Utilizados para alternar una única opción.

Reguladores

Permiten al usuario tener un control fluido sobre un valor (ajuste del volumen o configurar pantalla y brillo).

Campos de texto

Permiten al usuario ingresar una pequeña cantidad de texto.



Acciones y conectores

Puedes crear conectores que permitan a las acciones acceder a las propiedades de tus vistas y controles.

Reconocedores de gestos

Los gestos están vinculados a una sola vista, pero una vista puede tener varios gestos.

Auto Layout

Uso de restricciones para hacer interfaces de usuario adaptables.

Tipos de restricciones

- De alineación
- De tamaño
- De la pantalla
- De vistas similares





Si un botón está centrado de manera vertical y horizontal dentro de una vista, ¿Qué restricciones fueron las que con mayor probabilidad, se definieron?

- ¿Qué restricciones agregarías para colocar una vista en la esquina derecha inferior de una vista matriz?
- ¿Cuáles herramientas de Interface Builder puedes usar para crear restricciones?





En este tema conociste diferentes controladores que te ayudarán a ejecutar código a través de diferentes elementos, así como también a acceder al valor de una condición, usar campos de texto, responder a interacciones del usuario por medio de gestos reconocidos, entre otros.

Asimismo, conociste el uso del Auto Layout para construir vistas precisas y las restricciones que puedes crear.

