**PROGRAMACIÓN EN MÓVILES AVANZADOS : Consumir API desde una aplicación**

1. **Capacidades**

* Desarrollar e integrar APIs y servicios REST en aplicaciones iOS utilizando SwiftUI

1. **Seguridad**

* En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red. Así como la ingesta de alimentos y bebidas.
* Ubicar maletines y/o mochilas en lugar destinado para tal fin.
* Dejar la mesa de trabajo y la silla utilizada limpias y ordenadas.

1. **Fundamento teórico**

* Revise el material de la semana correspondiente antes del desarrollo del laboratorio.

1. **Normas empleadas**

* *No aplica*

1. **Recursos**

* En este laboratorio, cada estudiante trabajará con una computadora..

1. **Metodología para el desarrollo de la tarea**

* El desarrollo del laboratorio es en grupo de a 2 estudiantes

**VII.- PROCEDIMIENTO:**

**Actividad 01**

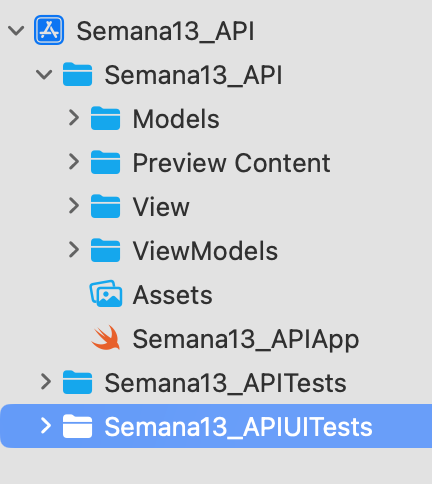
Realizar el consumo de una API con el objetivo de que muestre solo la lista de Usuarios.

Para ello debe de crear la siguiente la aplicación mediante SwiftUI y crear la siguiente estructura de carpetas , para realizarlo mediante MVVM

Direccion de la API a Consumir : <https://jsonplaceholder.typicode.com/users>

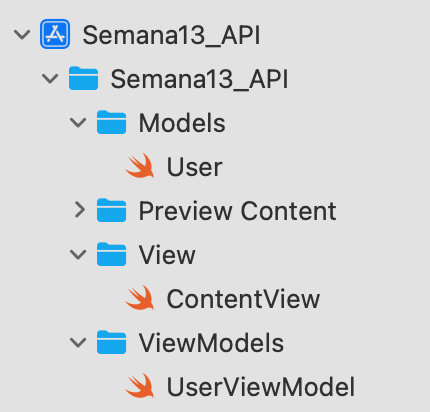
1. Crear la aplicación en SWiftUI

Seguidamente crear la siguiente estructura de Carpetas

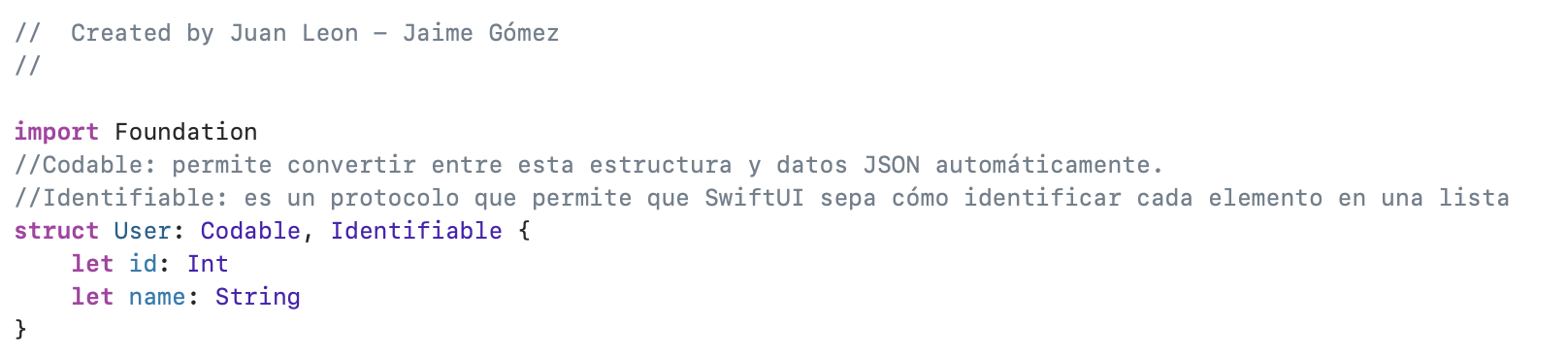


2. Arrastrar el archivo **ContentView** a la carpeta **View**  y crear los archivos :

**User** y **UserViewModel**  en las carpetas correspondientes según la imagen.

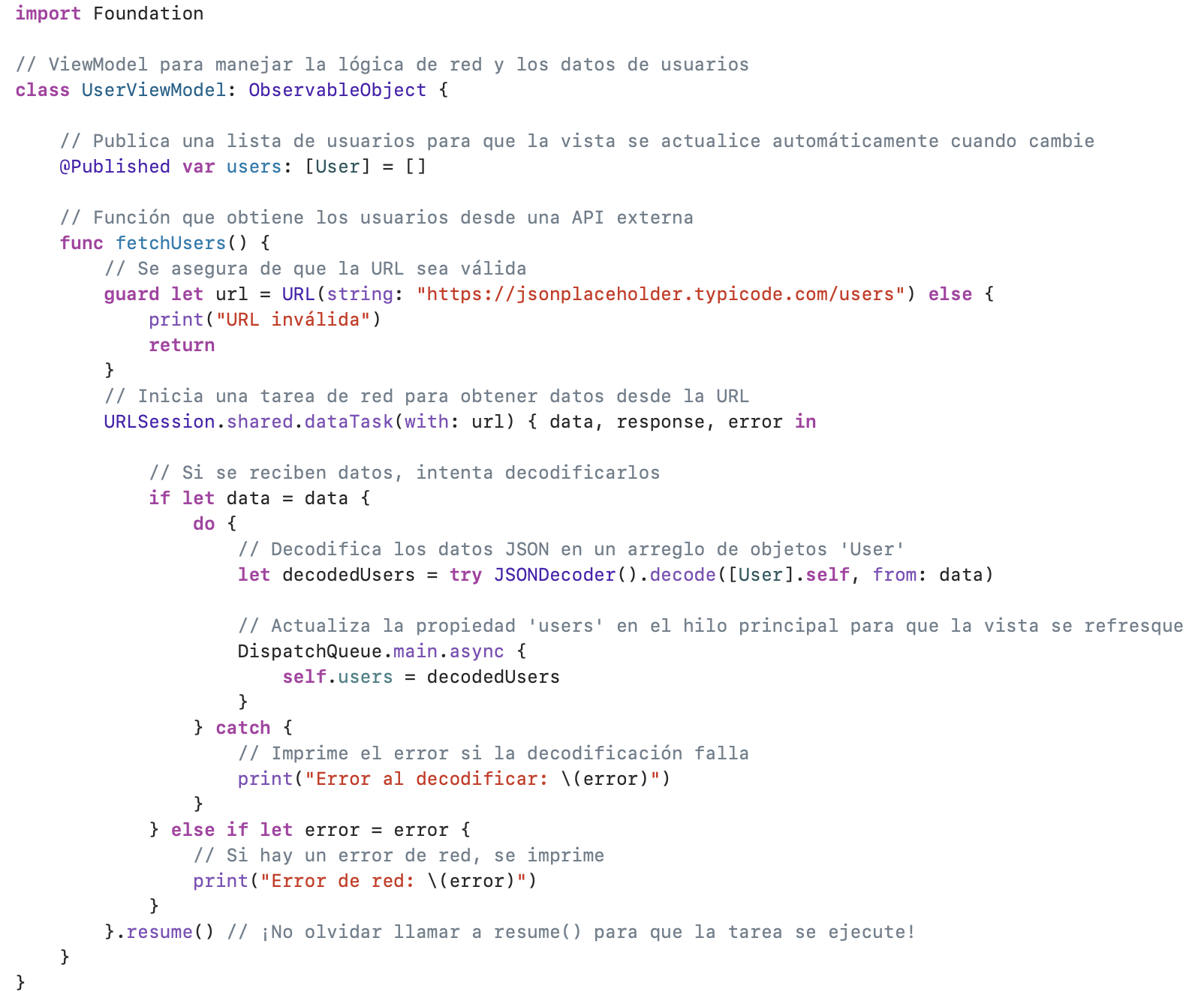


3. Contenido del Archivo **User**.

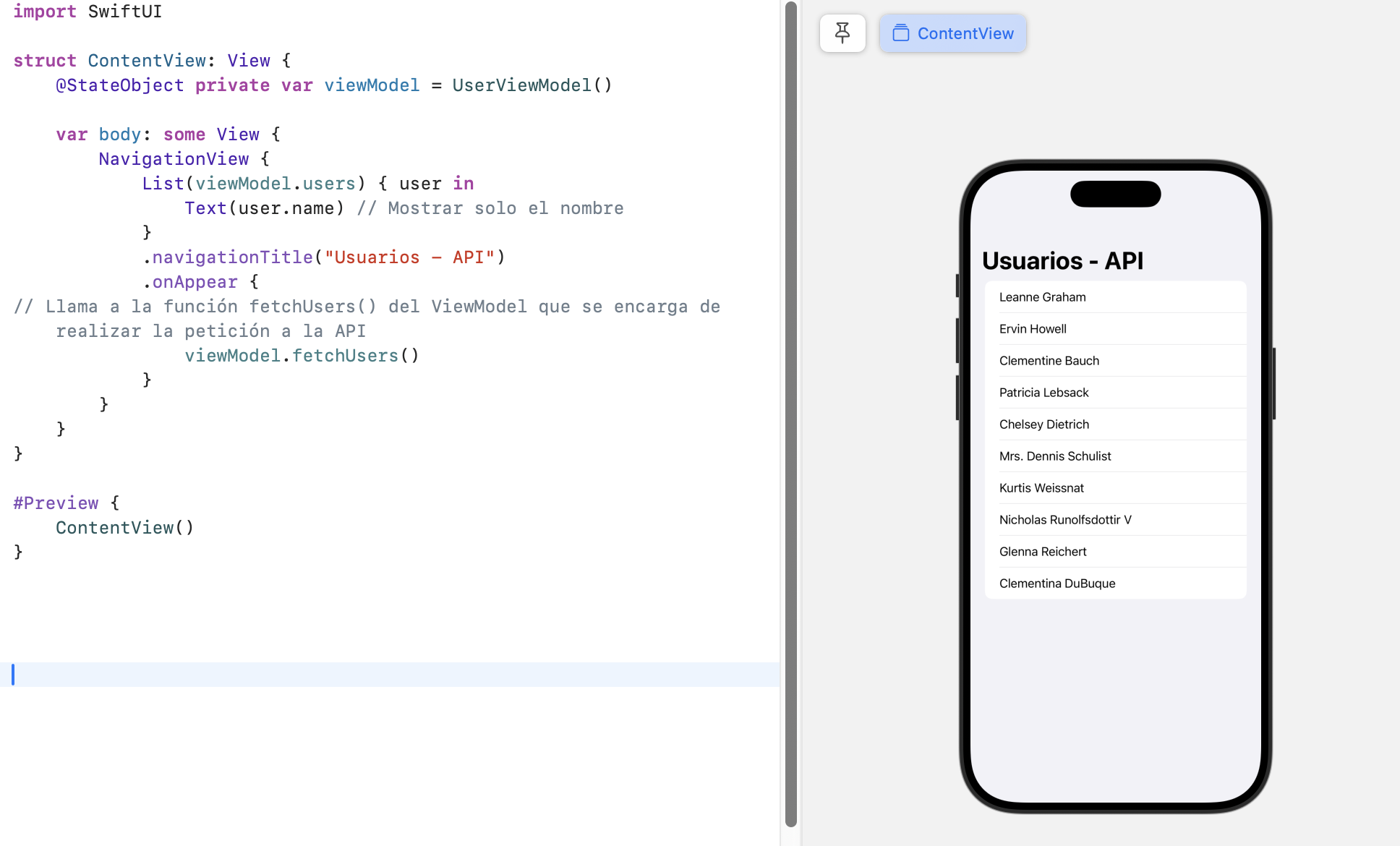


4. Contenido del archivo **UserViewModel**

En este archivo debe de utilizar la dirección de la API que se encuentra al inicio de la **Actividad1**



5. Contenido del Archivo Principal : **ContentView**

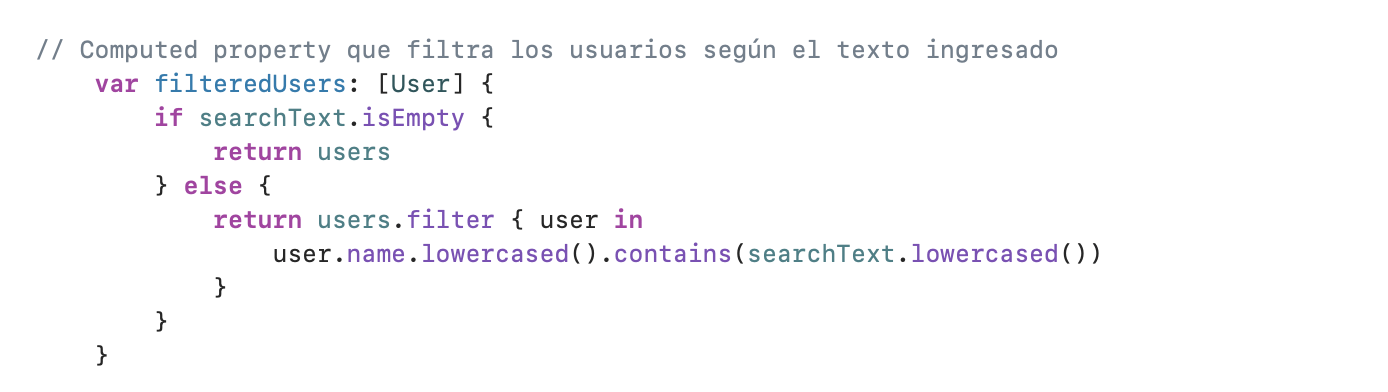


6. Después de Ejecutar la Aplicación vamos agregar un filtro de Búsqueda para ello solo vamos a modificar el archivo **UserViewModel**, agregando el siguiente código

Definiendo una nueva Variable

@Published **var** searchText: String = ""        // Texto del filtro

Tambien agregando el siguiente filtro de busqueda



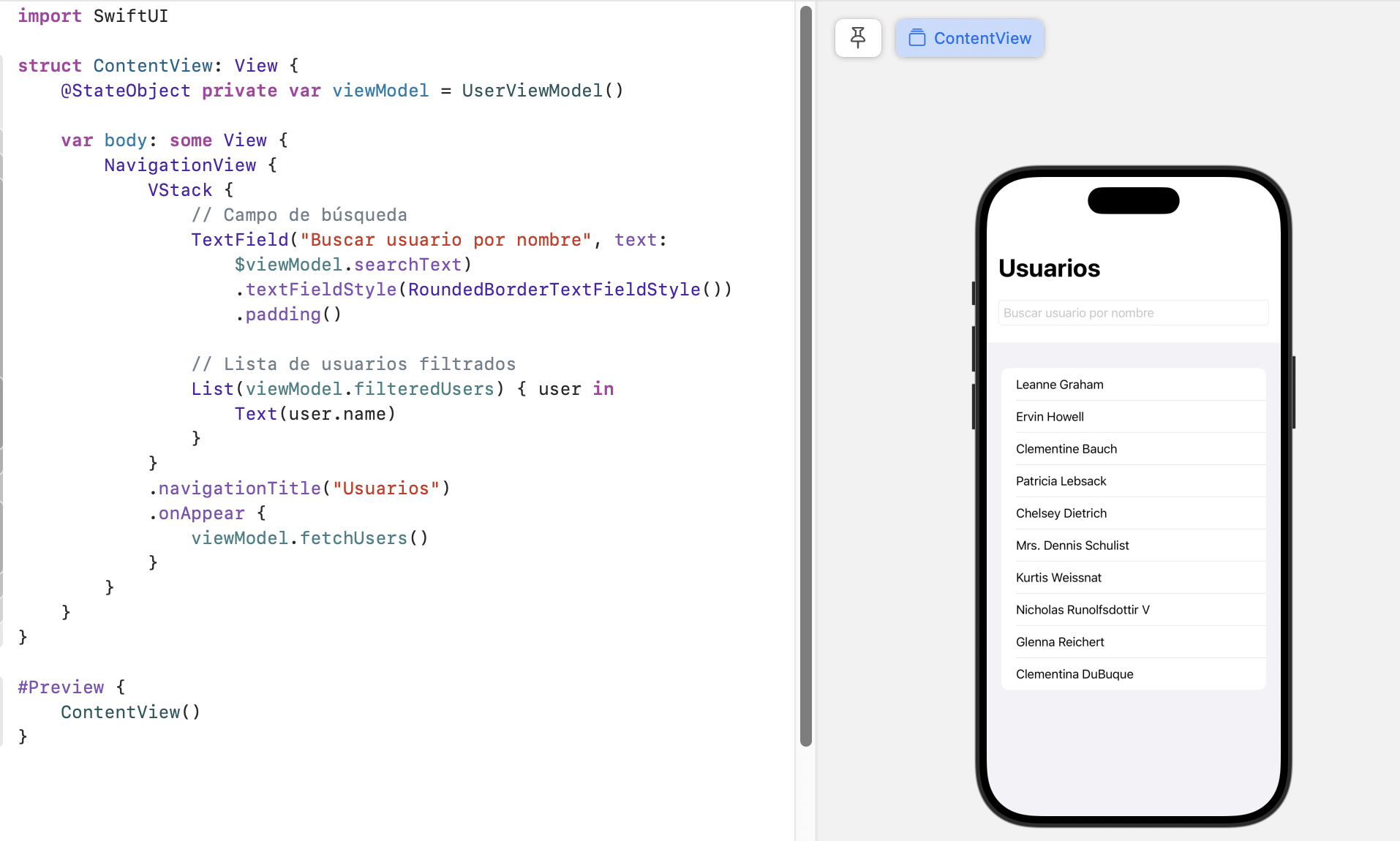
¿Cuál es la función del @Published?

|  |
| --- |

Explique brevemente que se realiza dentro de FiltereUsers

|  |
| --- |

7. Modificar también la vista principal del archivo ContentView:



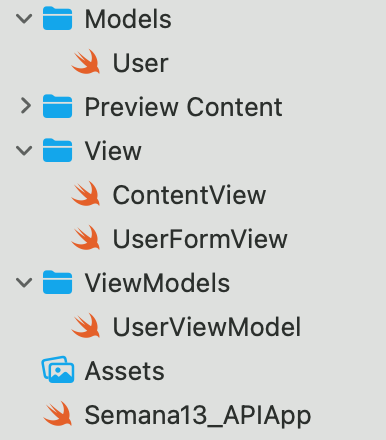
**Actividad 02**

Utilizar la API del ejercicio anterior para realizar los métodos POST PUT Y DELETE

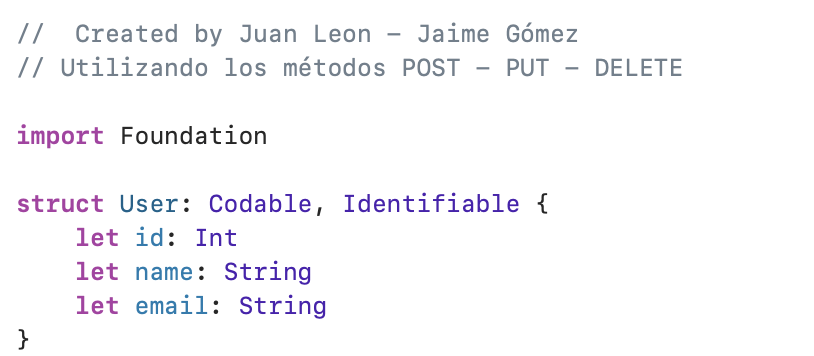
Nombre del Proyecto : Semana13\_MetodosAPI

Creando la misma estructura de la actividad anterior MVVM vamos a crear una aplicación ahora agregando las acciones de Agregar, Actualizar y eliminar.

Estructura similar al ejercicio anterior – Crear las carpetas con los nombres similares al ejercicio anterior y los archivos correspondientes en sus ubicaciones respectivas.



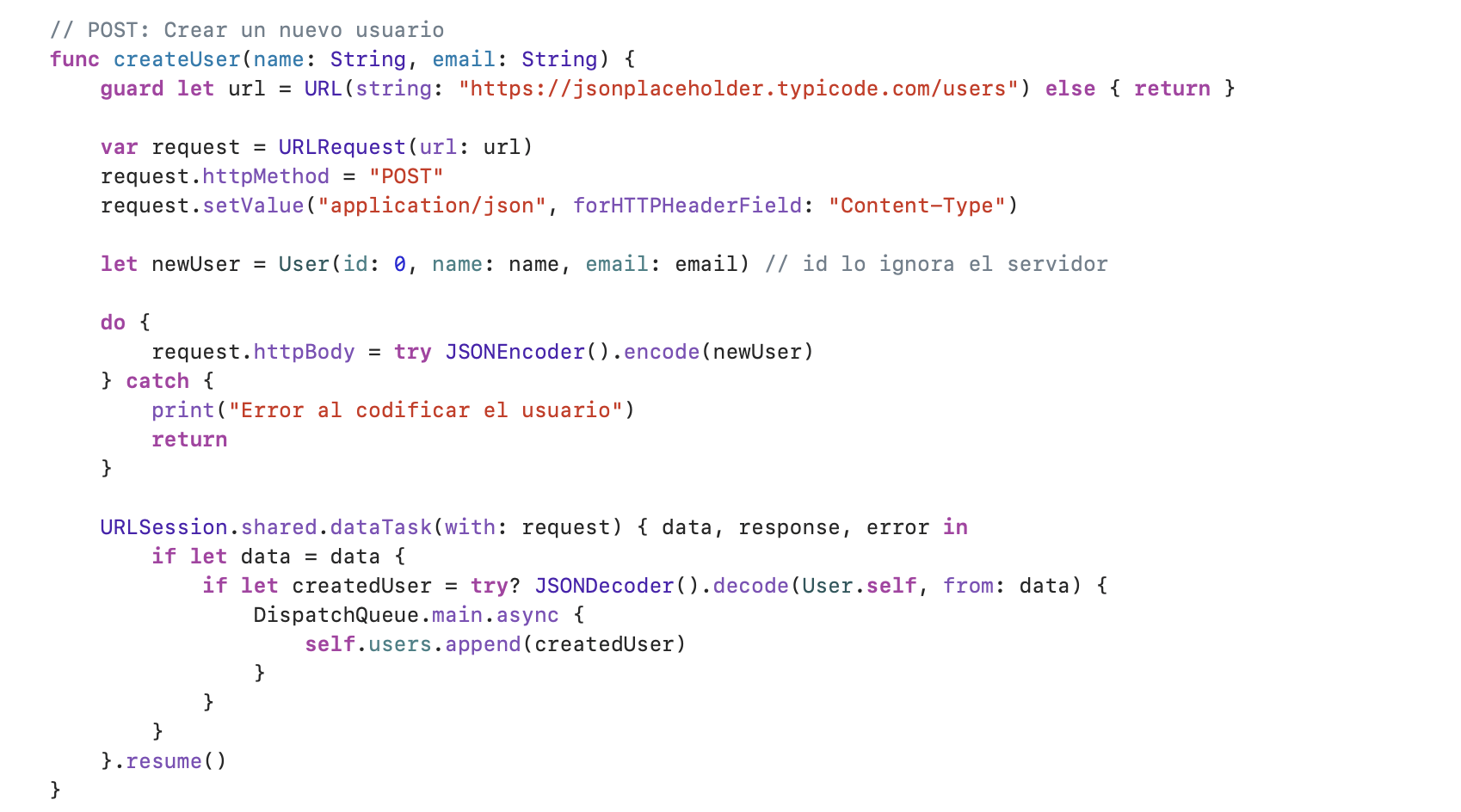
1. Crear el modelo en el archivo **User.**

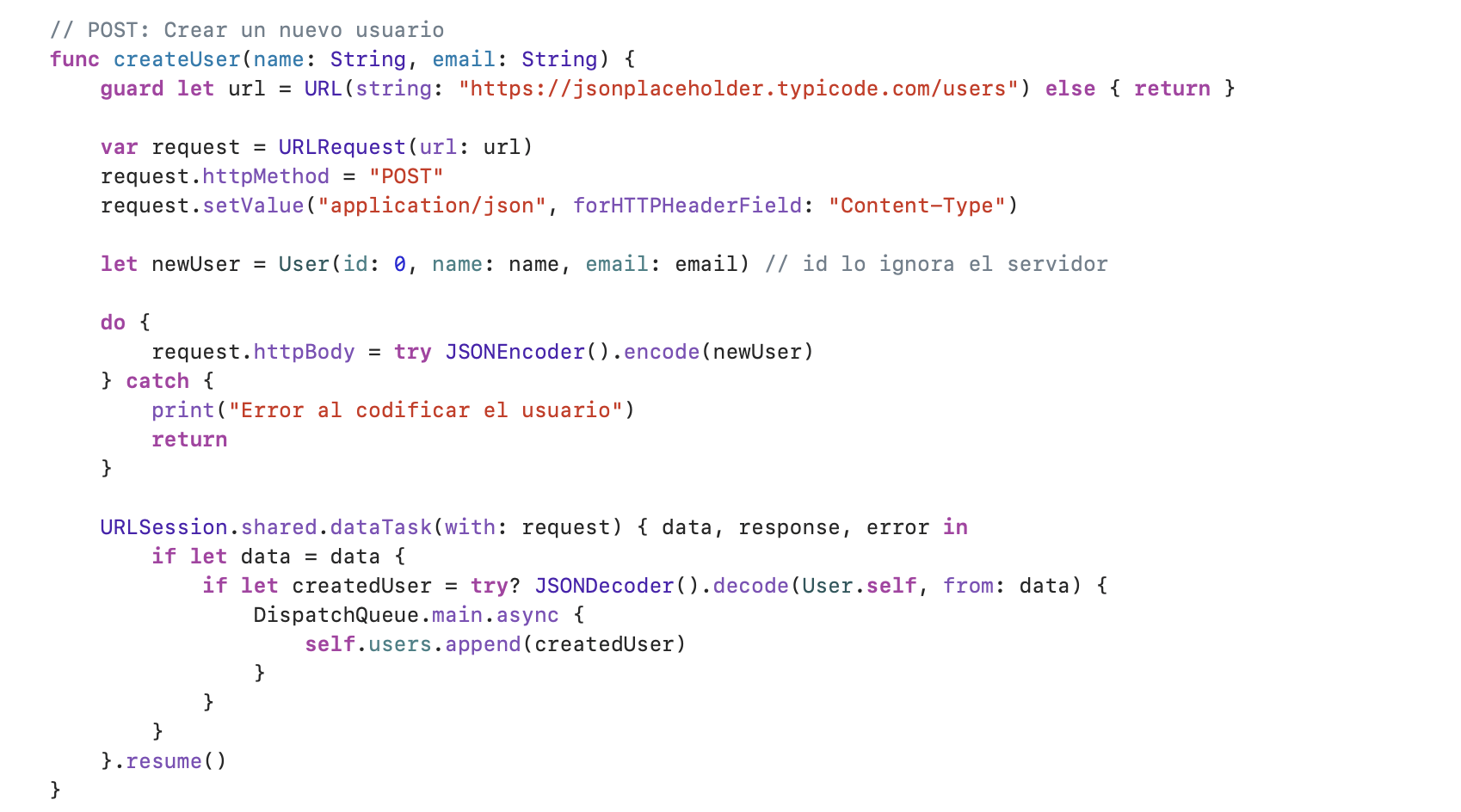


2. El Archivo **UserViewModel** donde se encuentra la lógica y el consumo de la API y donde se definen los métodos requeridos











¿Cuál es la finalidad de la función **resume() ?**

|  |
| --- |

¿Qué función cumple **DispatchQueue ?**

|  |
| --- |

¿Qué función cumple la clase JSONDecoder()?

|  |
| --- |

3. El archivo **UserFormView** el cual define la vista para agregar o modificar un usuario



4. El Archivo principal **ContentView** con algunas fuentes y colores para una mejor presentación





**Explicar las siguientes acciones**

 @State **private** **var** editingUser: User? = **nil**

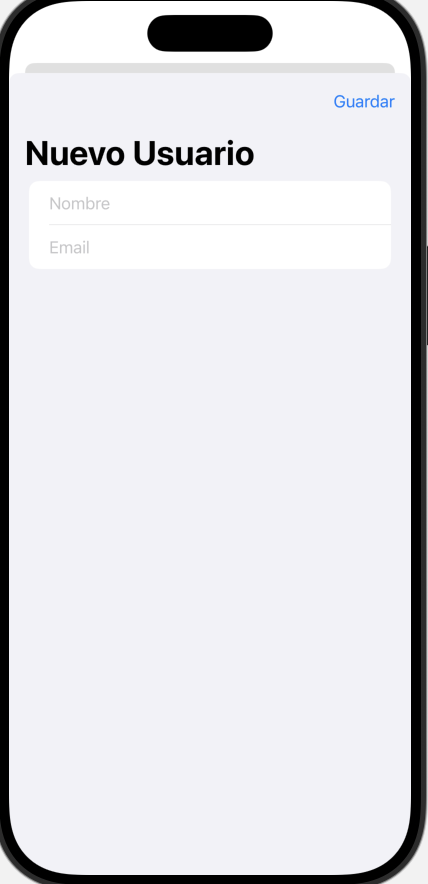
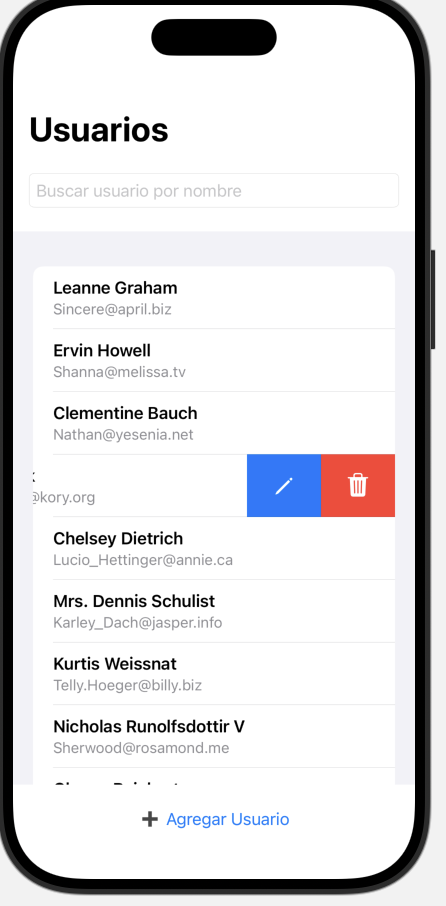
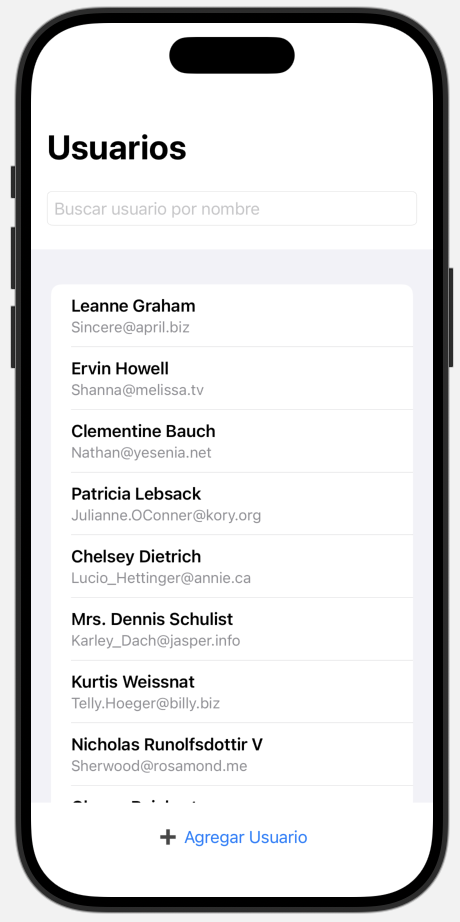
|  |
| --- |

@StateObject **private** **var** viewModel = UserViewModel()

|  |
| --- |

**.swipeActions**

|  |
| --- |

**.sheet(isPresented: $showingForm)**

|  |
| --- |

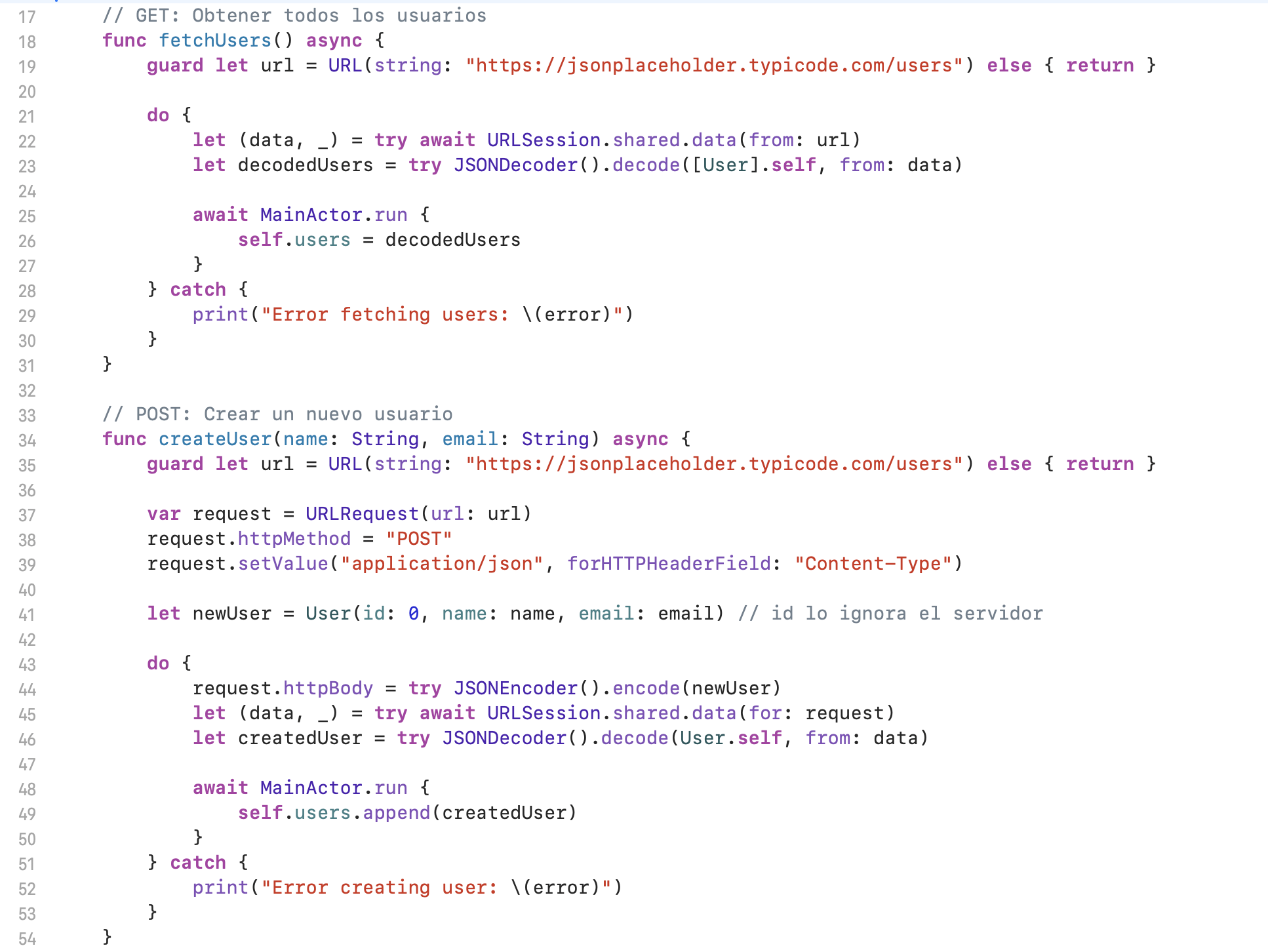
**Actividad 03**

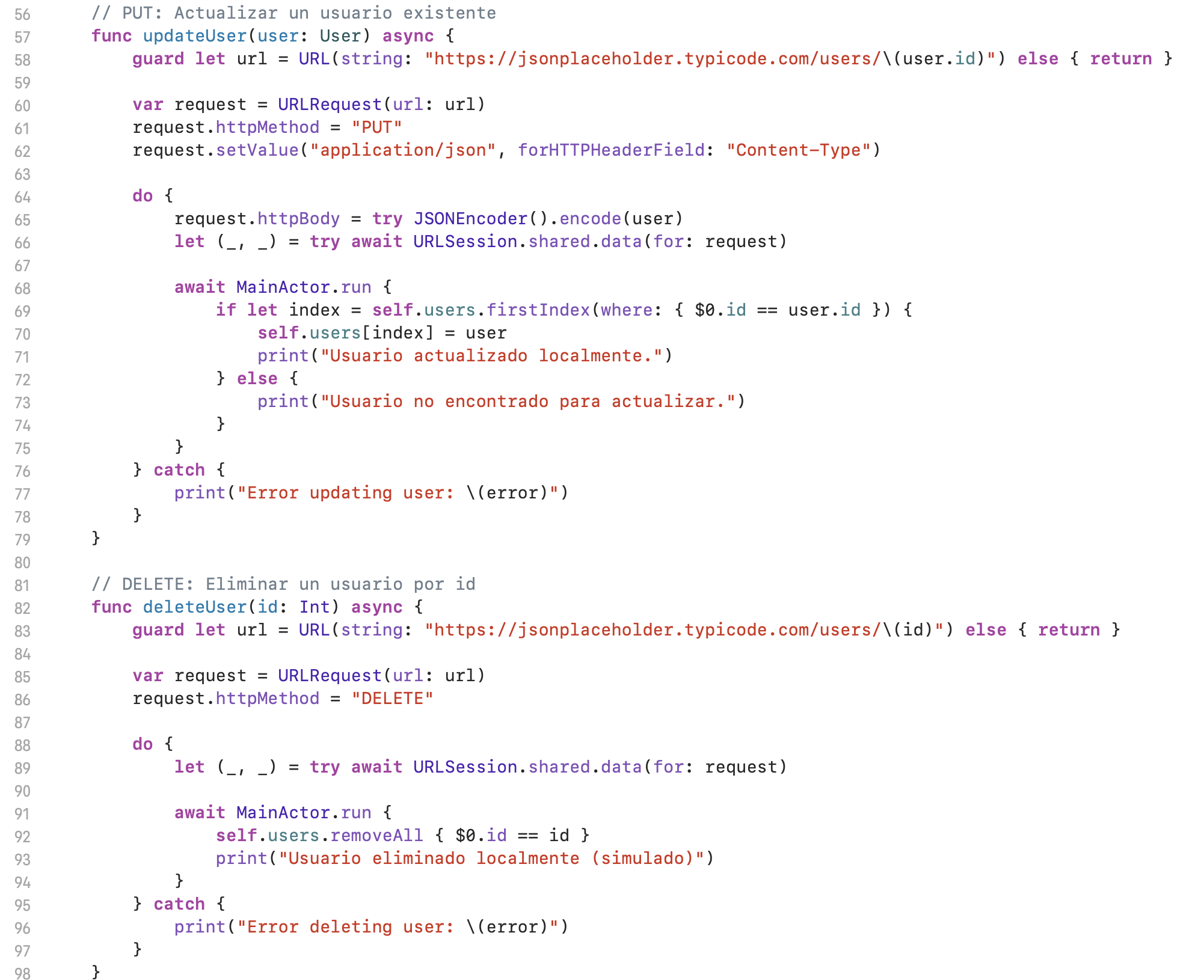
Uso de **URLSession** methods:

* URLSession.shared.data(from: url) for GET requests
* URLSession.shared.data(for: request) for POST, PUT, DELETE requests

Se va ha realizar cambios en las clases UserViewModel y ContentView

UserViewModel





ContentView

Los cambios están indicados en el rectángulo rojo



**Explicar cuál es la diferencia de usar el método URLSession.shared.data en vez de URLSession.shared.dataTask**

|  |
| --- |

**¿Para qué sirve la palabra reservada await ?**

|  |
| --- |

**¿Para qué sirve la estructura Task ?**

|  |
| --- |

**Conclusiones**

1. **…**
2. **…**
3. **…**