



# SNOWTRAILS

En un mundo donde la conexión con la naturaleza y la aventura son esenciales, nace **SnowTrails**, una aplicación diseñada para los amantes de las rutas de montaña.

SnowTrails pretende ser una plataforma intuitiva que permita a los usuarios acceder a información sobre rutas de montaña de manera fácil y rápida. Con ella, los senderistas podrán visualizar rutas y encontrar la manera de combinarlas para llegar de un punto a otro de rutas diferentes.

La misión de SnowTrails es transformar la manera en que los entusiastas del montañismo exploran sus rutas y ayudarlos a planificar su trayecto de manera segura y efectiva. Únete a nosotros en esta aventura y descubre todo lo que SnowTrails tiene para ofrecerte.

<b>1</b>	<b>Dataset</b>	<b>1</b>
<b>2</b>	<b>Funcionalidades obligatorias</b>	<b>2</b>
2.1	Menú de login . . . . .	2
2.2	Menú de usuario normal . . . . .	3
2.2.1	Visualización de las rutas . . . . .	3
2.2.2	Obtener ruta más corta . . . . .	4
2.2.3	Log out . . . . .	4
2.3	Menú de administrador . . . . .	4
2.3.1	Ver todos los usuarios . . . . .	4
2.3.2	Añadir usuario . . . . .	5
2.3.3	Eliminar usuario . . . . .	5
2.3.4	Añadir punto a una ruta . . . . .	5
2.3.5	Logout . . . . .	5
<b>3</b>	<b>Práctica complementaria</b>	<b>6</b>
3.1	Menú de login . . . . .	6
3.2	Menú de usuario normal . . . . .	6
3.2.1	Obtener ruta más corta . . . . .	6
3.3	Menú de administrador . . . . .	7
3.3.1	Añadir usuario . . . . .	7
3.3.2	Añadir punto a una ruta . . . . .	7
<b>4</b>	<b>Otros detalles de implementación</b>	<b>8</b>
<b>5</b>	<b>Entregables</b>	<b>8</b>

# 1 Dataset

Las rutas se forman por los siguientes puntos:

Table 1: Lista de puntos geográficos

Nombre	Latitud	Longitud	Elevación (m)
Alpina Grande	46.0000	7.5000	1500.0
Alpina Pequeña	46.0022	7.5200	1200.0
Pico Nevado	46.1000	7.6000	1600.0
Valle Blanco	45.9000	7.4000	1400.0
Cumbre Azul	46.0500	7.5500	1550.0
Lago Helado	46.2000	7.7000	1700.0
Bosque Nevado	46.3000	7.8000	1800.0
Cerro Plateado	46.1500	7.6500	1650.0
Cascada Blanca	46.2500	7.7500	1750.0
Refugio Alpino	46.0500	7.4500	1450.0
Refugio Aislado	46.0000	7.4000	1400.0

A su vez, estos puntos se conectan de manera natural entre ellos mediante estas conexiones:

Table 2: Conexiones entre puntos geográficos

Punto A	Punto B
Alpina Grande	Pico Nevado, Valle Blanco, Cumbre Azul, Alpina Pequeña
Alpina Pequeña	Alpina Grande
Pico Nevado	Alpina Grande, Lago Helado, Cerro Plateado
Valle Blanco	Alpina Grande, Refugio Alpino
Cumbre Azul	Alpina Grande, Cerro Plateado
Lago Helado	Pico Nevado, Bosque Nevado
Bosque Nevado	Lago Helado, Cascada Blanca, Cerro Plateado
Cerro Plateado	Pico Nevado, Cumbre Azul, Bosque Nevado, Cascada Blanca
Cascada Blanca	Cerro Plateado, Bosque Nevado
Refugio Alpino	Valle Blanco
Refugio Aislado	

Por último, estos puntos se han conectado de manera artificial en las rutas predefinidas que se muestran a continuación:

Table 3: Rutas y los puntos que contienen

Nombre de la ruta	Puntos
Ruta del Pico Nevado y Lago Helado	Alpina Grande, Pico Nevado, Lago Helado
Ruta del Valle Blanco y Refugio Alpino	Alpina Grande, Valle Blanco, Refugio Alpino
Ruta de la Cumbre Azul y Cerro Plateado	Alpina Grande, Cumbre Azul, Cerro Plateado
Ruta del Bosque Nevado y Cascada Blanca	Lago Helado, Bosque Nevado, Cascada Blanca
Ruta completa de Alpina Grande a Cascada Blanca	Alpina Grande, Pico Nevado, Lago Helado, Bosque Nevado, Cascada Blanca
Ruta del Refugio Aislado	Refugio Aislado
Ruta Alpina	Alpina Grande, Alpina Pequeña

## 2 Funcionalidades obligatorias

Esta sección contiene los **mínimos necesarios para aprobar el módulo**. Estas funcionalidades se dividen en tres niveles:

1. Menú de login
2. Menú de usuario normal
3. Menú de administrador

### 2.1 Menú de login

Se trata de una aplicación con control de acceso basado por roles en la que los usuarios deberán iniciar sesión, ya sea como **usuarios regulares** que buscan disfrutar de su tiempo en la nieve, o como **administradores** que controlan y gestionan la información de las rutas y usuarios. **No puede existir un usuario que no sea administrador o usuario normal**, pero todos comparten tres características: *nombre*, *email* y *contraseña*.

Cuando un usuario hace login, ha de poder **iniciar sesión** únicamente mediante su **email y contraseña**, que *se pedirán al usuario*. Una vez obtenidos estos credenciales, se deberá comprobar que existe en el registro de usuarios un nombre de usuario para dichos credenciales. En caso de no ser así, deberá *tratarse el error* correspondiente. Por descontado, **un usuario administrador no podrá acceder al menú de usuario regular con su cuenta de administrador y viceversa**.

Por defecto, deben existir un usuario regular y otro administrador cargados en el programa desde el inicio:

- Administrador
  - Nombre: Adminuserkeepcoding1
  - Email: adminuser@keepcoding.es
  - Contraseña: Adminuser1
- Usuario normal
  - Nombre: Regularuserkeepcoding1
  - Email: regularuser@keepcoding.es
  - Contraseña: Regularuser1

Por otra parte, es necesario darle un **feedback al usuario de que ha iniciado sesión, como usuario (normal) o como administrador**, antes de pasar al menú específico de cada tipo de usuario.

Por último, debe existir una **opción para salir de la aplicación** presente en este menú principal del login: cuando se selecciona esta opción, *el programa se cierra instantáneamente*.

A continuación, se presenta una *propuesta para la ejecución de este menú*:

Bienvenido a SnowTrails

1. Acceder como usuario
2. Acceder como administrador
3. Salir

## 2.2 Menú de usuario normal

El menú del usuario normal debe mostrar tres únicas opciones:

Menú usuario - Selecciona una opción:

1. Ver todas las rutas
2. Obtener la ruta más corta entre dos puntos
3. Log out

### 2.2.1 Visualización de las rutas

Si se selecciona la primera opción, se deben mostrar todas las rutas existentes con el formato *Nombre de la ruta - Distancia formateada con 2 decimales km* como en el siguiente ejemplo:

Ruta del Pico Nevado y Lago Helado - 27.07 km

Hay que tener en cuenta que si la ruta tiene **un único punto**, entonces la **distancia de la ruta será de 0.00 km** (en el dataset facilitado existe al menos una ruta con estas características).

**La distancia de las rutas** es un dato que **debe ser calculado** mediante la suma de las distancias de los puntos que las componen. Para ello, se deben considerar los puntos de las rutas en el orden en el que se indican en el dataset. En otras palabras, *el orden de los puntos en una ruta sí es importante*.

### 2.2.2 Obtener ruta más corta

En las funcionalidades obligatorias de la práctica final, únicamente se espera un **mensaje que indique que esta característica no está implementada** todavía para informar al usuario. Después del mensaje informativo, *se volverá a mostrar el menú de usuario normal*.

### 2.2.3 Log out

Esta opción **devolverá al usuario al menú de inicio de sesión**, en el que podrá elegir si hacer login como usuario normal o administrador. Además, debe mostrar un *mensaje por consola que indique que se está cerrando sesión*.

## 2.3 Menú de administrador

El menú de administrador debe mostrar las siguientes opciones:

Menú admin - Selecciona una opción:

1. Ver todos los usuarios
2. Añadir usuario
3. Eliminar usuario
4. Añadir punto a una ruta
5. Logout

### 2.3.1 Ver todos los usuarios

Cuando se seleccione la primera opción, **se mostrarán todos los usuarios** guardados en el programa con el siguiente formato:

Admin: Adminuserkeepcoding1 --- Email: adminuser@keepcoding.es

Regular user: Regularuserkeepcoding1 --- Email: regularuser@keepcoding.es

Esta opción **debe devolver al menos estos dos usuarios al inicio del programa**, ya que son los usuarios que deben cargarse por defecto.

### 2.3.2 Añadir usuario

El administrador debe poder **crear usuarios normales en el programa**. Para ello, el programa le **pedirá el nombre, email y contraseña** del nuevo usuario. *Al final, se mostrará un mensaje* para dar feedback al administrador de que la *creación del usuario ha tenido éxito*, con el nombre y el email del nuevo usuario:

```
Añadir usuario
Introduce el nombre del usuario que quieres añadir
...
Introduce el email del usuario que quieres añadir
...
Introduce la contraseña del usuario que quieres añadir
...
Usuario nombreDeUsuario con email emailDeUsuario añadido satisfactoriamente
```

Una vez creado, este usuario deberá aparecer si intentamos listar todos los usuarios. En caso de que ocurra algún error, deberá ser gestionado adecuadamente.

### 2.3.3 Eliminar usuario

Los usuarios normales pueden ser **eliminados mediante el nombre**. Si existen **dos usuarios con el mismo nombre, ambos serán eliminados**, pero *si no hay ningún usuario con el nombre introducido*, entonces se mostrará un *mensaje de error*.

### 2.3.4 Añadir punto a una ruta

En las funcionalidades obligatorias de la práctica final, únicamente se espera un **mensaje que indique que esta característica no está implementada** todavía para informar al usuario. Después del mensaje informativo, *se volverá a mostrar el menú del administrador*.

### 2.3.5 Logout

Esta opción **devolverá al usuario al menú de inicio de sesión**, en el que podrá elegir si hacer login como usuario normal o administrador. Además, debe mostrar un *mensaje por consola que indique que se está cerrando sesión*.

## 3 Práctica complementaria

### 3.1 Menú de login

Tanto el **usuario normal** como el **administrador** debe contar con **una manera de obtener el rol de usuario como una cadena de texto**. Esto debe ser una funcionalidad presente en todos los usuarios, pero el comportamiento ha de ser específico de cada tipo de usuario. *Si no es posible obtener el rol del usuario, debe lanzarse un error con un mensaje que lo indique.*

Existen ciertos requisitos para los datos de los usuarios:

- Email: debe tener la forma xxxxxx@xxxxxx.zz donde xxxxxx puede ser cualquier carácter, mayúscula, minúscula o número y zz puede ser la terminación .es o .com.
- Nombre: debe ser una cadena de texto de entre 8 y 24 caracteres que pueden ser letras mayúsculas, minúsculas y números.

Si todo va bien, el resultado **exitoso** deberá gestionarse en una closure **onSuccess**, pero si alguna de estas restricciones **falla**, debe tratarse el error consecuentemente mediante una closure **onFailure**. Asimismo, en caso de que el login haya sido satisfactorio, se deberá *imprimir por pantalla al usuario que se está accediendo como administrador o como usuario* (como usuario normal).

### 3.2 Menú de usuario normal

#### 3.2.1 Obtener ruta más corta

Al seleccionar la opción de obtener la ruta más corta entre dos puntos, se mostrarán las rutas y los puntos que las forman con el siguiente formato:

Ruta del Pico Nevado y Lago Helado con los siguientes puntos:

Alpina Grande  
Pico Nevado  
Lago Helado

De esta manera, el usuario podrá **copiar el nombre de cualquier punto**, que **será el punto de inicio** de la ruta más corta. Una vez introducido el primer punto, *se mostrarán de nuevo todas las rutas y sus puntos* para que el usuario vuelva a copiar e introducir cualquiera de los puntos.

En caso de que **los dos puntos introducidos existan y sean diferentes**, el programa debe calcular *la ruta más corta desde el primer punto hasta el segundo*. Es importante que se indique la distancia total de esta ruta más corta, así como la lista de los puntos que la componen con el *formato presentado a continuación* (ejemplo desde Alpina Grande hasta Cascada Blanca):



La distancia más corta entre Alpina Grande y Cascada Blanca es: 60.91 km.  
La ruta es:

Alpina Grande  
Cumbre Azul  
Cerro Plateado  
Cascada Blanca

Si se produjera **algún error** en el proceso, **deberá ser gestionado** para evitar que el programa deje de funcionar. Cabe destacar que *si se elige el mismo punto de inicio y fin*, el output será un *mensaje de error* en el que se informe al usuario de que los dos puntos deben ser diferentes.

### 3.3 Menú de administrador

#### 3.3.1 Añadir usuario

Las mismas restricciones que se aplican al login de los usuarios deben gestionarse en la creación de usuarios:

- Email: debe tener la forma xxxxxx@xxxxxx.zz donde xxxxxx puede ser cualquier carácter, mayúscula, minúscula o número y zz puede ser la terminación .es o .com.
- Nombre: debe ser una cadena de texto de entre 8 y 24 caracteres que pueden ser letras mayúsculas, minúsculas y números.

No obstante, **añadimos otra restricción más: el usuario no debe existir previamente en el programa**, aunque pueden existir usuarios con diferente nombre pero con un mismo email.

#### 3.3.2 Añadir punto a una ruta

Este punto de la práctica **depende de cómo se hayan implementado las rutas y los puntos que contienen**. Por ello, el input esperado dependerá de los datos que requiera la implementación.

En cualquier caso, **una vez añadido un punto a una ruta, debe poder utilizarse como uno de los puntos iniciales o finales para calcular la ruta más corta**. Asimismo, *la ruta de la que forma parte debe actualizarse con la información correspondiente*.

## 4 Otros detalles de implementación

El programa tendrá **dos tipos de mensajes**, **logs para el desarrollador** que le indiquen qué está ocurriendo en las entrañas del programa y **mensajes dirigidos al usuario final** del programa. Por ello debe existir una *entidad que se encargue de imprimir mensajes por pantalla hacia el usuario*.

De igual manera, en el programa tiene que haber un **controlador del menú** que se tiene que encargar de todos los menús, un controlador del login que se encarga del inicio de sesión, un controlador de las rutas, etc. **La responsabilidad de estos controladores será la de interactuar con el usuario a través del controlador del menú**. También son válidas otras implementaciones, siempre que existan entidades exclusivas de interacción con el usuario.

En cuanto a los datos, todas las conversiones, manipulaciones y demás **operaciones que no tengan relación con la interacción con el usuario** tienen que estar alojados en otros componentes. Es bastante común encontrar estas funcionalidades en clases o structs denominados **Services**. De este modo, tendríamos un UserService que se encargue de los usuarios, un RoutesService cuya responsabilidad son las rutas, etc.

Por último, es necesario que exista una **entidad App que sea la que inicie la aplicación**. Por lo tanto, nuestro punto de entrada en el programa será el siguiente código del archivo main.swift:

```
let app = App()
app.run()
```

## 5 Entregables

- Fecha de entrega: **16/02/2025 a las 24:00 hora española**
- El proyecto se subirá a github con los siguientes requisitos:
  - Proyecto con el **código fuente** que compile
  - Código documentado en la medida de lo posible
  - **Testing** a partir del 30 - 40 de porcentaje de coverage está bien
  - **Readme**
    - \* Explicación general del proyecto
    - \* Especificaciones concretas que sea conveniente señalar
    - \* Indicación de los puntos del enunciado que hayáis completado
    - \* Comentarios sobre la práctica, dudas no resueltas, dificultades y demás consideraciones