

# **TPO: PROGRAMACIÓN III**

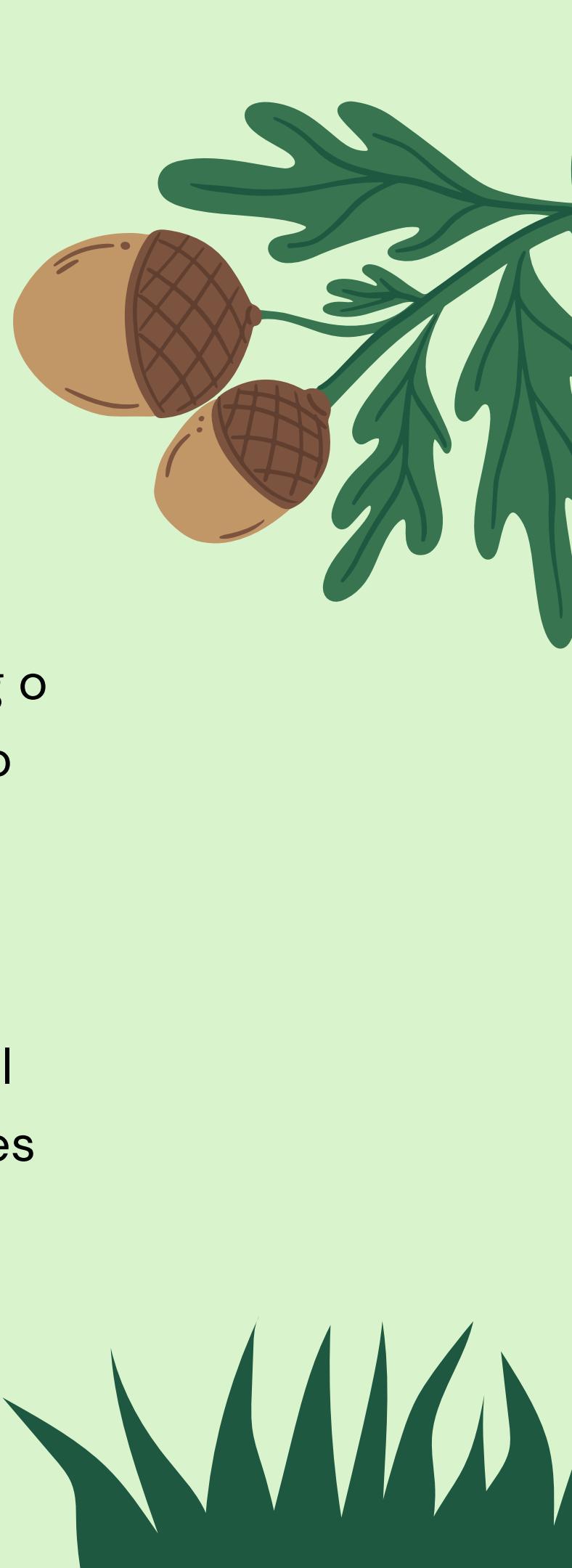
# **PROYECTO TREKKING**

Grupo 8

Gael Kugler - Ariana Progano Elmallian



# **¿EN QUÉ CONSISTE NUESTRO PROYECTO?**



La idea del sistema es para esos aventureros que deciden hacer trekking o senderismo, los caminos usuales de trekking tienen ya un mapa armado donde se puede saber de forma anticipada que se encuentra por cada camino o lugares que te ayudan a saber en qué parte estas.

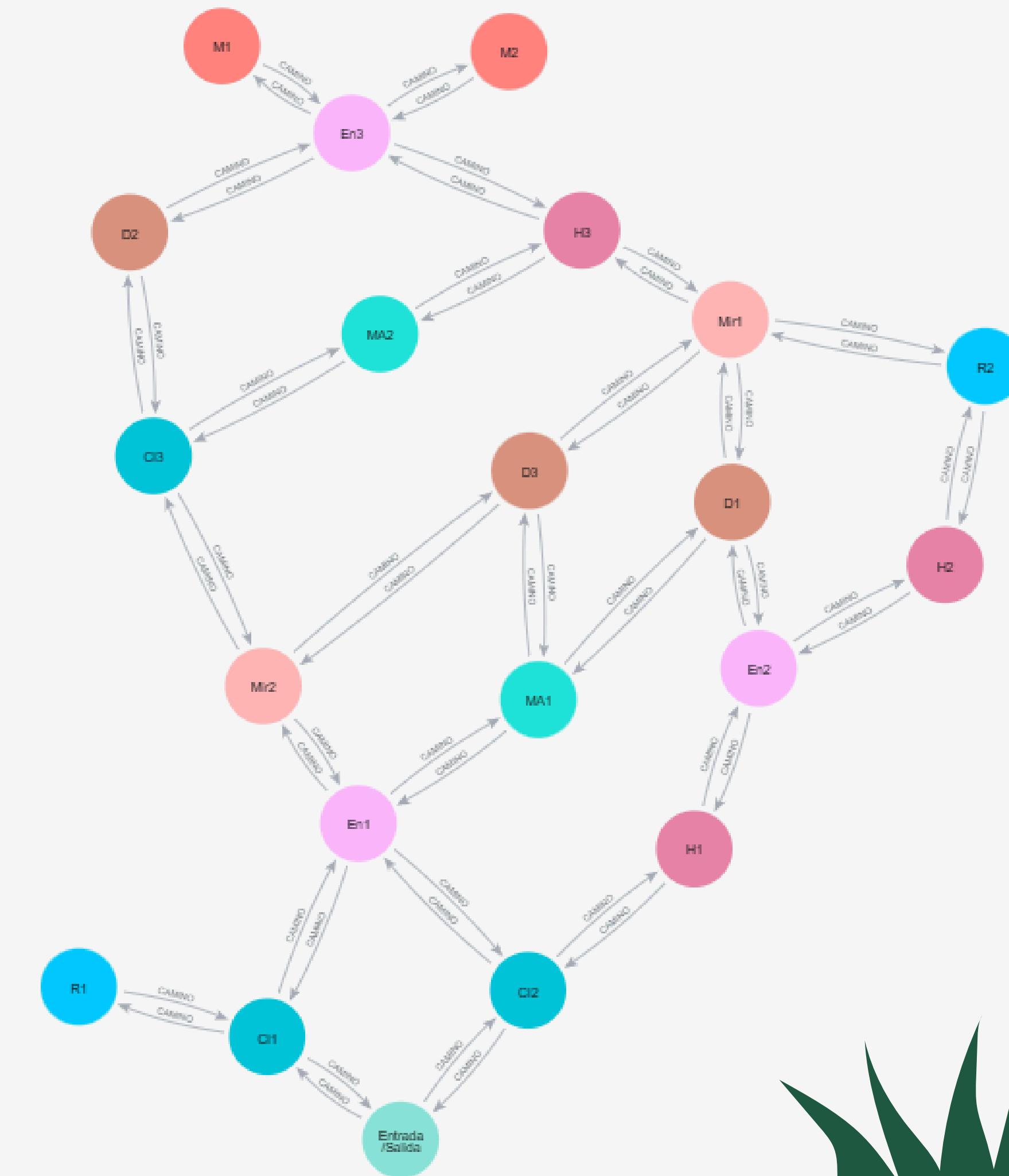
Un usuario tiene acceso a un mapa del parque y a partir de ahí, según el sendero que quiera hacer a través de distintos algoritmos muestra cuáles son las rutas posibles o la mejor ruta.

# NUESTRO GRAFO

Cada nodo es un punto identificador:

- Zona de descanso (D)
- Mirador (Mir)
- Entrada y Salida (E/S)
- Cartel Informativo (CI)
- Puesto de Hidratación (H)
- Montaña (M)
- Rio (R)
- Mirador Aves (MA)
- Encrucijadas (En)

Las aristas son los caminos naturales entre esos puntos.



# ALGORITMOS UTILIZADOS

BFS Y DFS

GREEDY

DIJKSTRA

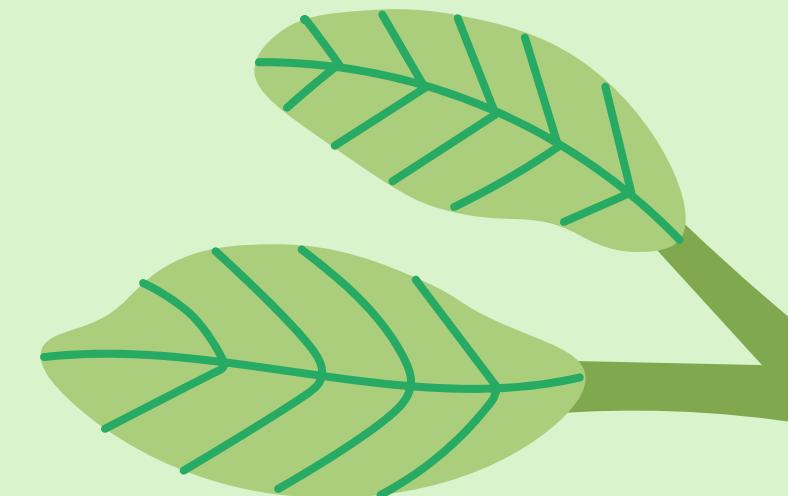
PROGRAMACIÓN DINÁMICA

PRIM

BACKTRACKING

KRUSKAL

BRANCH & BOUND



# EJEMPLOS DE APLICACIÓN

## GREDDY VS PROGRAMACIÓN DINAMICA

El usuario tiene poco tiempo y quiere aprovecharlo visitando la mayor cantidad de lugares posibles. Hay un límite de km, el usuario va a poder elegirlo.



# EJEMPLOS DE APLICACIÓN



## DIJKSTRA

"Me dijeron que la Montaña 1 es lo mas maravilloso del recorrido pero estoy medio cansado por lo tanto, me gustaria saber cual es el recorrido mas corto en cuestión de distancia para poder llegar a la Montaña 1"



## BFS VS DFS

"Estoy en el Mirador 1 y necesito llegar a la zona de descanso 2. Quisiera saber qué lugares o puntos identificadores tengo que atravesar para llegar a la zona de descanso más cercana"



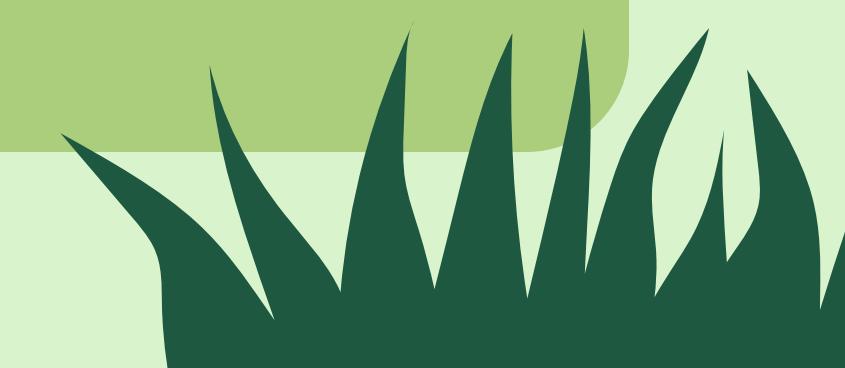
# EJEMPLOS DE APLICACIÓN

## PRIM

Un usuario quiere realizar el trekking pero se da cuenta que tiene las piernas cansadas, así que decide quedarse en la zona de descanso, no tiene problemas en tardar algunos días en recorrer todo al 100% pero considerando la zona de descanso como su refugio donde siempre vuelve.

## KRUSKAL

Un administrador del sendero se da cuenta que al haber tantos caminos distintos, es mucho trabajo cuidarlos a todos o hay algunos que nunca se utilizan siquiera ya que el público general solo desea visitar los miradores e irse, no quieren recorrer cada camino largo.



# EJEMPLOS DE APLICACIÓN

## BACTRACKING

El usuario decide previamente como quiere realizar su recorrido, donde indica que solo quiere ir a una montaña y tomar máximo 1 una zona de descanso. Backtracking nos daria las distintas rutas donde se cumplen estas restricciones.

## BRANCH & BOUND

“Estoy en la zona de descanso 2 y ya me quiero ir, pero antes me gustaría poder visitar el Rio 2, como ya pude descansar, no quiero pasar por las zonas de descanso por nada en el mundo, por lo tanto quiero saber ¿Cual es la mejor ruta que cumpla con lo que pido?”



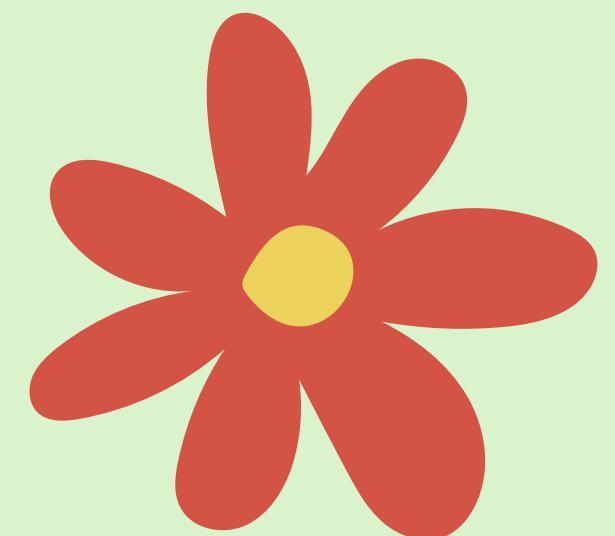


# ¿POR QUÉ NO LO APLICAMOS?

## DIVIDE Y VENCERÁS

Decidimos no implementar divide y vencerás ya que son algoritmos que consisten en dividir una lista en 2 y organizar los elementos ya sea con quicksort (puntero) o mergesort (comparando pares).

No nos parece que haya algun caso donde el usuario desearia ordenar una lista donde implemente este algoritmo y que le sea de utilidad.



**¡MUCHAS  
GRACIAS!**