

En la definición del tipo `hospital_t` agrego un `size_t` con la cantidad de entrenadores ya que es necesario para implementar la función `hospital_cantidad_entrenadores`.

En las funciones `hospital_cantidad_entrenadores`, `hospital_cantidad_pokemon`, `pokemon_nivel` y `pokemon_nombre` tan solo retorno el valor solicitado.

Para la función `crear_hospital` reservo memoria para un hospital e inicializo sus variables, reservando memoria para un vector de pokémones.

Contrariamente en la función `destruir_hospital` recorro todo el vector de pokémones liberando la memoria reservada para sus nombres, luego libero el vector de pokémones y finalmente la memoria utilizada para el hospital en sí.

Para la función `hospital_leer_archivo` creo dos funciones auxiliares que leen las líneas del archivo abierto y luego las agregan al hospital. La función que agrega los datos leídos al hospital se encarga de reservar la memoria necesaria para dichos datos y modifica la cantidad de pokémones si fuera necesario.

Para la función `hospital_a_cada_pokemon` implemento una función de ordenamiento alfabético mediante bubble sort y luego llamo a la función que recibí con el vector `pokemon` hasta que devuelva false. Retorno la cantidad de pokémones recorridos al momento que la función es false.