* **¿Cuándo hacer una lista y cuándo hacer un vector?**
  + Vector: Cuando sabés exactamente cuántos datos tenés que guardar.
  + Lista: Cuando no sabés cuántos datos vas a guardar.
* **¿Cuándo uso un vector contador?**
  + Generalmente los usás cuando piden calcular cantidades de ciertos tipos (por ejemplo, si el enunciado te dice que los datos se separan en 5 categorías, y después te piden calcular cuántos productos hay por categoría. En ese caso el vector seria de 5 posiciones).
  + Acordate que los vectores contadores SIEMPRE se inicializan en 0 ANTES DE USARLO.
* **¿Cuándo uso parámetros por valor y cuándo por referencia?**
  + Por valor: cuando no vas a modificar el contenido y solo te interesa saber el valor que guarda esa variable.
  + Por referencia: cuando querés modificar el contenido y que se vea desde donde se lo llamó al módulo.
* **¿Cuándo uso una función?**
  + Las funciones se usan cuando el módulo devuelve un tipo de dato SIMPLE (punteros, reales, enteros, booleanos).
  + Recordá que los registros y los strings no son tipos de datos simples.
  + En las funciones NO SE DEBE ni leer ni escribir (read y write).
* **¿Cuándo “uso corte de control”?**
  + Generalmente los algoritmos de corte de control se pueden usar cuando los datos vienen ordenados bajo cierto criterio.
* **¿Cuándo modularizo?**
  + Hay que pensar que un módulo (ya sea función o procedimiento) debe ser lo más genérico y reutilizable posible. Lo ideal es que solo se encargue de hacer una tarea en especifico. Si el módulo que hiciste hace mas de una tarea, entonces se puede seguir modularizando.
* **¿Vale la pena modularizar una suma?**
  + Si esas pequeñas líneas de código las usas muchas veces a lo largo de tu programa, entonces tal vez podrías modularizarlas para que quede mas legible el código final.
* **¿Qué debo escribir cuándo el enunciado me dice “se dispone”?**
  + Lo único que hay que hacer cuando ya se dispone de una estructura es escribir el type completo y en el programa hacer un llamado ej: “cargarLista(L);” y nada más. No hay que implementar ningún modulo ni nada.
  + Recordar poner un comentario al lado del llamado de la carga diciendo “se dispone”.
* **¿Qué cambia que el enunciado diga que cuando tengo que borrar un elemento de la lista puede que no exista o seguro existe?**
  + Si el elemento puede no existir: hay que preguntar por act <> nil, tanto en el while que recorre como en el if que elimina
  + Si el elemento seguro existe: no hace falta preguntar por act <> nil ni en el while ni en el if.