

Pasos para resolver parcial a mano:

- |  |        |
|--|--------|
| 1. Leer el enunciado todas las veces necesarias        | 15 min |
| 2. Plantear idea general                               | 10 min |
| 3. Definir los struct                                  | 10 min |
| 1. regArch   |        |
| 2. infoLS  |        |
| 3. nodoLS  |        |
| 4. infoLP  |        |
| 5. nodoLP  |        |
| 4. Definir las estructuras de datos                    | 10 min |
| 1. Constantes  |        |
| 2. Archivos  |        |
| 3. Vectores  |        |
| 4. Listas  |        |
| 5. Dejar espacio para tipo de dato de func y prototipo |        |
| 6. Codificar función                                   | 40 min |
| 1. Pensar funcionamiento general                       |        |
| 2. Definir variables                                   |        |
| i. auxArch   |        |
| ii. auxReg   |        |
| iii. auxNodo   |        |
| iv. auxIndice  |        |
| v. regSize   |        |
| vi. contador   |        |
| vii. resultado   |        |
| 3. Precondiciones para el while                        |        |
| 4. While y condicionales                               |        |
| 7. Codificar prototipo de función                      | 5 min  |

Acordarse de las funciones de biblioteca y sus prototipos

Abreviar variables

### Precondiciones:

- Búsqueda binaria: Ordenado e incompleto
- Apareo: Ordenados
- While: Lectura anticipada
- InsertarOrdenado: Inicializar en NULL
- Contar en vec: Inicializar en 0
- Lista de Lista: nodoAux, auxLS, auxLP
- Mostara Lista: auxLista para no perder

### Prototipos:

- int BusquedaBinaria(reg vec[], int tLogico, reg buscado) // Campo Clave...
- void Burbuja(reg vec[], int tLogico) // Criterio de Ordenamiento...
- nodo\* InsertarOrdenado(**nodo\* &lista**, reg val) // Criterio de Ordenamiento
- void Push(**nodo\* &pila**, reg val)
- reg Pop(**nodo\* &pila**)
- void Queue(**nodo\* &fte**, **nodo\* &fin**, reg val)
- reg UnQueue(**nodo\* &fte**, **nodo\* &fin**)